



# Proyecto Final

Ingeniería en Sistemas de Información 2021



## Gestión de Eventos Deportivos

**Facultad:** FRM UTN

**Carrera:** Ing. en sistemas de información

**Año:** 2021

**Curso:** 5K10

**Docentes:**

- Vazquez, Alejandro
- Moralejo, Raúl
- Manino, Gustavo

**Integrantes:**

- Ignacio Starkloff (44068)
- Miguel Del Corso (44052)
- Juan Segundo Riera (43774)
- José Aranciba (44161)
- Elvio Aruta (43988)

# Índice de contenidos

<b>Índice de contenidos</b>	<b>2</b>
<b>Índice de figuras</b>	<b>20</b>
<b>Índice de anexos</b>	<b>20</b>
<b>Trabajo Práctico 1</b>	<b>21</b>
Resumen.	22
Palabras Clave.	22
Conclusiones.	22
Definición de requerimientos.	24
Eventos	24
1) Relevamiento general.	24
1.1) De la organización.	24
1.2) Funciones detectadas.	24
1.3) Tecnología de información.	26
2) Relevamiento detallado y análisis del Sistema.	27
2.1) Detalle, explicación y documentación detallada de todas las funciones seleccionadas.	27
2.2) Modelo lógico del Sistema actual.	34
2.3) Problemas y necesidades detectados en las funciones relevadas en detalle y en su entorno organizacional.	34
SportEasy	35
1) Relevamiento general.	35
1.1) De la organización.	35
1.2) Funciones detectadas.	35
1.3) Tecnología de información.	36
2) Relevamiento detallado y análisis del Sistema.	37
2.1) Detalle, explicación y documentación detallada de todas las funciones seleccionadas.	37
2.2) Modelo Lógico del sistema actual.	40
EventBase	41
1) Relevamiento general.	41
1.1) De la organización.	41
1.2) Tecnología de Información.	42
2) Relevamiento detallado y análisis del Sistema.	42
2.1) Detalle, explicación y documentación detallada de todas las funciones seleccionadas.	42
2.2) Modelo lógico del Sistema actual.	45
Diagrama de contexto.	47
2.3) Problemas y necesidades detectados en las funciones relevadas en detalle y en su entorno organizacional.	47

Playinga	48
1) Relevamiento general	48
1.1) De la organización	48
1.2) Funciones detectadas e interfaces	48
1.3) Tecnología de Información	49
2) Relevamiento detallado y análisis del Sistema.	49
2.1) Detalle, explicación y documentación detallada de todas las funciones seleccionadas.	49
2.2) Modelo lógico del Sistema actual.	57
2.3) Problemas y necesidades detectados en las funciones relevadas en detalle y en su entorno organizacional.	58
SportsPlus	58
1) Relevamiento general	58
1.1) De la organización	58
1.2) Funciones detectadas	58
1.3) Tecnología de información	59
2) Relevamiento detallado y análisis del sistema.	60
2.1) Detalle, explicación y documentación detallada de todas las funciones seleccionadas.	60
2.2) Modelo lógico del sistema actual.	66
2.3) Problemas y necesidades detectados en las funciones relevadas en detalle y en su entorno organizacional.	66
2.3 Cuadro resumen de las funciones detectadas en los sistemas relevados	67
2.4 Objetivos y alcances preliminares del nuevo Sistema.	68
2.4.1 Objetivos preliminares del sistema:	68
2.4.2 Tecnologías a utilizar en el desarrollo	69
2.4.3 Alcances preliminares del nuevo sistema	69
Módulo de seguridad	69
Módulo de eventos	70
Módulo de reportes	71
Módulo de publicidad	71
Módulo de inscripción	71
Módulo de pagos	71
Módulo de Feedback	71
Módulo de parametrización	72
Módulo Perfil	72
Diseño	74
1. Objetivos y alcances definitivos del nuevo Sistema.	74
Objetivo del Sistema	74
Alcance Funcional	74
Módulo de seguridad	74

Módulo de eventos	76
Alta Evento	76
Modificación Evento	77
Eliminar Evento	77
Interactuar con el Evento	77
Módulo de reportes	78
Módulo de publicidad	80
Módulo de inscripción	81
Inscribirse a un evento	81
Darse de baja a un evento	81
Módulo de pagos	82
Pago digital	83
Pago en efectivo	83
Módulo de Feedback	84
Dar de alta un feedback	84
Dar de baja un feedback	85
Módulo de parametrización	86
Dar de alta un parámetro	86
BackUp	87
Módulo Recomendación	88
Recibir las recomendaciones	88
Parámetros usados en las recomendaciones	88
Módulo Sharing	89
Compartir un elemento	90
Compartir un evento luego de crearlo	90
Módulo Perfil	91
2. Salidas del Sistema.	93
Módulo Seguridad	93
Módulo Eventos	97
Módulo de inscripción	104
Módulo de Feedback	105
Módulo de sharing	106
Módulo de reportes	107
Módulo de Parametrización	109
Módulo pagos	118
Módulo Publicidad	119
Módulo Perfil	120
3. Modelo funcional	128
Historias de Usuario	128
Formato Historia de Usuario	128
Especificación de historias de usuario	130
Módulo Seguridad	130

Módulo de eventos	142
Módulo de recomendación	158
Módulo de inscripción	159
Módulo de sharing	165
Módulo de perfil	169
Módulo Parametrización	190
Módulo de reporte	211
Módulo de feedback	213
Módulo de Pago	214
Módulo de publicidad	218
4. Pantallas	218
5. Modelo de datos	299
Diagramas de transición de estados	300
<b>Trabajo Práctico 2</b>	<b>305</b>
<b>CAPÍTULO I: Actividades</b>	<b>306</b>
1.1 Definición y descripción de actividades.	306
1.1.0 Definición del sistema	306
1.1.1 Relevamiento General	306
Relevamiento de las organizaciones	306
Funciones detectadas	306
Tecnologías de Información	307
1.1.2 Relevamiento detallado y análisis del Sistema.	307
Detalle, explicación y documentación detallada de todas las funciones seleccionadas	307
Modelo lógico del sistema	307
Problemas y necesidades detectados en las funciones relevadas en detalle y en su entorno organizacional	307
Objetivos y alcances preliminares del nuevo Sistema	307
1.1.3 Planificación	307
Estimación de actividades	308
Selección de metodologías	308
Definición del equipo de trabajo	308
Definición de tecnologías	308
Definición de herramientas de comunicación	308
Definición de herramientas para el seguimiento de tareas	308
Definición de herramientas para la gestión de configuración del software	309
Definición y descripción de recursos para cada una de las actividades	309
Realizar estudio de factibilidad.	309
Costos de infraestructura y tecnologías	309
Análisis de riesgos	309
Análisis de impacto ambiental	309

1.1.4 Diseño	309
Objetivos y alcances definitivos del nuevo Sistema.	309
Salidas del sistema	309
Mockups	310
Modelo de datos	310
Definición de casos de uso	310
Definición de historias de usuario	310
Estimación de historias de usuario	310
Definición de Ceremonias de la metodología ágil	310
Diseño de módulo seguridad	311
Diseño de módulo eventos	311
Diseño de módulo reportes	311
Diseño de módulo publicidad	311
Diseño de módulo inscripción	311
Diseño de módulo pagos	312
Diseño de módulo feedback	312
Diseño de módulo parametrización	312
Diseño de módulo perfil	312
Diseño de pruebas de integración	312
Diseño de pruebas de seguridad	312
Diseño de pruebas de stress/carga	313
1.1.5 Desarrollo e implementación	313
Desarrollo del formato sobre el cual vamos a trabajar	313
Programación, pruebas y documentación de los módulos	313
Realizar el manual de usuario del Sistema completo	313
Implementación del Sistema.	313
Realizar capacitación a usuarios	313
Demo del sistema y ensayo de exposición	314
Exposición anual de Proyecto de Sistemas	314
1.2 Diagrama de tiempos	314
<b>CAPÍTULO II: Organización para la ejecución del proyecto.</b>	<b>315</b>
2.1 Equipo de trabajo	315
2.1.1 Metodología a adoptar por el equipo de trabajo	315
2.1.2 Tecnologías a usar por el equipo de trabajo	315
Angular	315
Redux	316
Bootstrap	316
El stack tecnológico de backend está compuesto por:	317
Node.js	317
Express.js	317
Kubernetes	318

Docker	318
MongoDB	319
Redis	319
Mongoose	320
Cloudinary	320
VS code	321
WebStorm	321
El stack tecnológico de pruebas está compuesto por:	322
Jmeter:	322
Jasmine:	322
2.1.3 Puestos de trabajo y cantidad de personas por puesto	323
2.1.4 Descripción de los puestos de trabajo	323
Desarrollador Frontend:	323
Desarrollador Back-End:	323
Ingeniero QA:	324
DevOps:	324
Analista:	324
Diseñador:	325
Scrum Master:	325
Product owner:	325
DBA:	325
2.1.5 Perfiles para los puestos de trabajo	326
Desarrollador Frontend:	326
Desarrollador Back-End:	326
Ingeniero QA:	327
DevOps:	327
Analista:	328
Diseñador:	328
Scrum Master:	329
Product owner:	329
DBA:	329
2.1.6 Estructura del equipo de trabajo	330
2.2 Funciones principales de los miembros del equipo	330
2.3 Métodos de comunicación formal, control de avance, retroalimentación, decisiones.	331
2.3.1 WhatsApp	331
2.3.2 Discord	331
2.3.2 Google drive	332
2.3.3 Moodle	332
2.3.4 Trello	333
2.4 Gestión de Configuración del Software: Método de gestión de versionado durante todo el proyecto.	333

2.4.1 Gestión de configuración del software	333
2.4.2 Formato del versionado	334
<b>Capítulo III: Factibilidad</b>	<b>335</b>
3.1 Definición y descripción de recursos para cada una de las actividades.	335
3.2 Diagrama de recursos	339
3.3 Análisis de factibilidad	344
3.3.1 Factibilidad Técnica	344
Ventajas y desventajas de las tecnologías utilizadas en el proyecto	346
GIT	346
MONGODB	346
ANGULAR	347
NODEJS	347
KUBERNETES	348
DOCKER	348
WHATSAPP	348
DISCORD	349
GOOGLE DRIVE	350
MOODLE	350
TRELLO	351
Conclusiones de factibilidad técnica	352
3.3.2 Factibilidad Operativa	352
Conclusiones de factibilidad operativa	354
3.3.3 Factibilidad Legal	354
Conclusiones de la factibilidad legal	354
3.3.4 Factibilidad Económica	355
3.3.4.1. Costos económicos generales iniciales	355
Retorno de inversión	355
3.3.5. Beneficios económicos intangibles	355
3.3.6. Costos económicos intangibles	356
Conclusiones de la factibilidad económica	356
3.4 Costos desagregados por recursos (personal, tecnología) con periodicidad mensual	356
3.5 Análisis de riesgos	362
Definición de Riesgos	362
3.6 Análisis de impacto ambiental	365
Consumo de internet	366
Consumo de papel	366
Consumo de energía	366
Difusión de eventos	367
Acceso a eventos	367
Centralización	367



Adaptabilidad	367
Conclusión del análisis de impacto ambiental	368
Conclusión del análisis.	368
<b>Desarrollo e Implementación</b>	<b>369</b>
Programación y documentación	370
Tecnologías utilizadas	370
El stack tecnológico de gestión de la configuración está compuesto por:	370
Gitlab	370
El stack tecnológico de pruebas está compuesto por:	371
Jmeter:	371
Jasmine	371
El stack tecnológico de frontend está compuesto por:	372
Angular	372
Redux	372
Bootstrap	373
El stack tecnológico de backend está compuesto por:	373
Node.js	373
Express.js	374
Docker	374
Kubernetes	375
MongoDB	375
Redis	376
Mongoose	376
Cloudinary	377
vs code	377
WebStorm	377
Estructura del proyecto en Visual Code	378
Frontend	378
Backend	380
Estilo de codificación	381
Organización para el desarrollo	382
Documentación módulo de seguridad	383
HU N°1 SignUp Deportista	383
Planilla de documentación	383
Documentación del BackEnd	385
Pantallas	390
Documentación del FrontEnd	393
HU N°2 SignUp Club	404
Planilla de documentación	404
Documentación del BackEnd	407
Pantallas	407

Documentación del FrontEnd	410
HU N°3 SignUp Patrocinador	420
Planilla de documentación	420
Documentación del BackEnd	421
Pantallas	422
Documentación del FrontEnd	423
HU N°4 SignUp Organizador	425
Planilla de documentación	425
Documentación del BackEnd	427
Pantalla	427
Documentación del FrontEnd	428
HU N°5 SignUp Árbitro	429
Planilla de documentación	432
Documentación del BackEnd	433
Pantallas	433
Documentación del FrontEnd	435
HU N°6 LogIn	437
Planilla de documentación	439
Documentación del BackEnd	441
Pantallas	445
Documentación del FrontEnd	447
HU N°7 Log Out	451
Planilla de documentación	451
Documentación del BackEnd	452
Pantallas	453
Documentación del FrontEnd	454
HU N°8 Cambiar Email	456
Planilla de documentación	456
Documentación del BackEnd	458
Pantallas	459
Documentación del FrontEnd	460
HU N° 9 Cambiar Contraseña	467
Planilla de documentación	467
Documentación del BackEnd	469
Pantallas	469
Documentación del FrontEnd	470
HU N°10 Cambiar Contraseña desde email	477
Planilla de documentación	477
Documentación del BackEnd	479
Pantalla	481
Documentación del FrontEnd	482
HU N° 11 Contactarse con el administrador del sistema	484

Planilla de documentación	484
Pantalla	487
Documentación FrontEnd	488
Estructura de la documentación de backend	493
Readme Principal	493
Endpoint de cada Módulo	494
Planificación de capacitación	495
Introducción	495
Capacitación para los usuarios	495
Objetivos de la capacitación	495
Destinatarios	496
Temas de capacitación	496
Recursos a emplear	496
Estrategias o modalidades a utilizar	496
Cronograma de capacitación	497
Capacitación para el administrador del sistema	497
Planificación, ejecución y documentación de pruebas	498
Planeación de pruebas	498
Alcance	498
Fuera de alcance	498
Roles y responsabilidades	498
Metodología de prueba	499
Creación de datos de prueba	499
Estructura general de los casos de prueba	499
Duración de las pruebas	500
Actividades	500
Complejidad del test	501
Entregables	501
Herramientas de testing	501
Recursos del entorno de pruebas	501
Pruebas Unitarias	501
Pruebas de Seguridad	505
Pruebas de Integración	514
Prueba de integración entre módulos seguridad y sharing	524
Pruebas de Carga	526
Pruebas de Estrés	530
Manual de usuarios	534
Guionado de videos de manual de usuario	534
Landing Page	534
Acceder al sistema	534
Cerrar Sesión	535
Contactarse con el administrador	535

Eventos	535
Visualizar eventos	535
Inscripción a eventos	536
Perfiles	536
Editando el perfil del usuario	537
Funciones especiales de cada rol	539
Visualización de los deportes	539
Visualización de las categorías	540
Visualización de equipos	540
Visualización y creación de roles	540
Visualización y creación de países	540
Publicidad	540
Reportes	540
Descripción de los listados o reportes (con datos) con un ejemplo de cada uno y la descripción de cómo hay que hacer para obtenerlos.	541
Normas necesarias	542
Planificación de Implementación del Sistema	544
Objetivos de la implementación del Sistema	544
Personas involucradas en la implementación	544
Temas del Plan de Implementación	545
Instalación de servidores	545
Equipos de conectividad, comunicaciones, configuración, infraestructura, configuración de servidores	545
Método de conversión elegido	545
Conocimientos requeridos	546
Requerimientos técnicos	546
Cronograma de la Implementación	546
Preparación y creación de un clúster de Kubernetes con estándares de niveles productivos	547
Asignación de DNS y configuración del TLS para producción.	547
Creación y despliegue de una base de datos que soporte niveles productivos.	549
Carga inicial de datos	549
Migración	549
Creación de un usuario administrador	550
Verificación que las funcionalidades básicas funcionen adecuadamente.	550
Elaborar plan de recuperación	550
Implementación de backup	550
Recuperación del backup	551
Implementación	552
Implementación con okteto	552
Puestos de trabajo	568
Métodos de replicación y replicación de implementación en otros sitios	569

Capacitación del administrador 569

**Bibliografía 569**

## Índice de figuras

Figura 1: Logo Eventols	24
Figura 2: Diagrama de contexto de inscripción online de Eventols	25
Figura 3: Diagrama de contexto de pagos online de Eventols	26
Figura 4: Landing page de Eventols	27
Figura 5: Datos principales en Eventols	28
Figura 6: Alta de un evento en Eventols	28
Figura 7: Alta de un evento en Eventols	28
Figura 8: Configuración de las competencias en Eventols	29
Figura 9: Pantalla principal del módulo de análisis de Eventols	29
Figura 10: Interfaz del módulo de reportes de Eventols	30
Figura 11: Interfaz del módulo de promociones de Eventols	30
Figura 12: Interfaz del módulo de promociones de Eventols	31
Figura 13: Interfaz del módulo de inscripción de Eventols	31
Figura 14: Interfaz del módulo de pagos de Eventols	32
Figura 15: Interfaz del módulo de acreditar de Eventols	32
Figura 16: Interfaz del módulo de acreditar de Eventols.	32
Figura 17: Interfaz del módulo de inscripción de Eventols.	33
Figura 18: Modelo de casos de uso de Eventols	34
Figura 19: Logo SportEasy	35
Figura 20: Perfil del deportista en SportEasy	38
Figura 21: Alineación del equipo en SportEasy	38
Figura 22: Interfaz de resultados de SportEasy	39
Figura 23: Interfaz del módulo de estadísticas de SportEasy	40
Figura 24: Interfaz del módulo de mensajería de SportEasy	40
Figura 25: Modelo de casos de uso de SportEasy	41
Figura 26: Logo eventbase	41
Figura 27: Logo Join In	42
Figura 28: Interfaz del módulo de gamificación de EventBase	43
Figura 29: Interfaz del módulo de navegación de EventBase	44
Figura 30: Interfaz del módulo de agenda con los eventos próximos de EventBase	45
Figura 31: Modelo de casos de uso de EventBase	46
Figura 32: Diagrama de contexto para el módulo de pagos de EventBase	47
Figura 33: Diagrama de contexto para el módulo de localización de EventBase	47
Figura 34: Logo Playinga	48
Figura 35: Pantalla para la creación del equipo	50

Figura 36: Pantalla para la organización partido	51
Figura 37: Pantalla del partido de playinga	51
Figura 38: Interfaz de comentarios de playinga	52
Figura 39: Interfaz de creación de un torneo en playinga	53
Figura 40: Interfaz de creación de un torneo en playinga	54
Figura 41: Interfaz para poder personalizar el “branding” del equipo	54
Figura 42: Interfaz del módulo de pagos de playinga	55
Figura 43: Facturación en playinga	56
Figura 44: Retiro de un jugador en playinga	57
Figura 45: Módulo de casos de uso de playinga	57
Figura 46: Logo SportsPlus	58
Figura 47: Interfaz de inicio de sesión de SportsPlus	60
Figura 48: Módulo de seguridad de SportsPlus	61
Figura 49: Perfil de SportsPlus	62
Figura 50: Calendario en SportsPlus	63
Figura 51: Módulo de notificaciones de SportsPlus.	64
Figura 52: Perfil del equipo de SportsPlus.	65
Figura 53: Módulo de información del evento de SportsPlus	66
Figura 54: Diagrama de casos de uso de SportsPlus	66
Figura 55: Diagrama de contexto del módulo de seguridad	76
Figura 56: Diagrama de contexto del módulo de eventos	78
Figura 57: Diagrama de contexto del módulo de reportes	80
Figura 58: Diagrama de contexto del módulo de inscripción	82
Figura 59: Diagrama de secuencia API Mercado Pago	83
Figura 60: Diagrama de contexto del módulo de pagos	84
Figura 61: Diagrama de contexto del módulo de feedback	86
Figura 62: Diagrama de contexto del módulo parametrización	88
Figura 63: Diagrama de contexto del módulo de recomendaciones	89
Figura 64: Diagrama de contexto del módulo de sharing	91
Figura 65: Pantalla inicial de registro	219
Figura 66: Pantalla de registro de deportista	219
Figura 67: Pantalla de registro de patrocinador	220
Figura 68: Pantalla de registro de club	221
Figura 69: Pantalla de registro de organizador	222
Figura 70: Pantalla de registro de árbitro	223
Figura 71: Pantalla de inicio de sesión	224
Figura 72: Pantalla de recuperación de contraseña	224
Figura 73: Pantalla 1 de creación de evento	225
Figura 74: Pantalla 2 de creación de evento	226
Figura 75: Pantalla de modificación de evento	226
Figura 76: Pantalla de eliminación de evento	227
Figura 77: Pantalla de visualización de resultados de partidos de jornada	227

Figura 78: Pantalla de visualización de jornada como organizador	228
Figura 79: Pantalla de registro de resultado de partido	228
Figura 80: Pantalla de creación de equipo como deportista	229
Figura 81: Pantalla de modificación de equipo	229
Figura 82: Pantalla de visualización de solicitudes de eventos como club	230
Figura 83: Pantalla para aceptar/rechazar petición evento como Club	231
Figura 84: Pantalla de visualización de ofertas de patrocinio como Organizador	232
Figura 85: Pantalla para aceptar/rechazar oferta de patrocinio como Organizador	233
Figura 86: Pantalla de visualización de evento como deportista	234
Figura 87: Pantalla de selección de categoría a la que se quiere inscribir en evento	235
Figura 88: Pantalla de selección de equipo con el que se quiere inscribir al evento	236
Figura 89: Pantalla de pago de inscripción	237
Figura 90: Pantalla para visualizar evento	238
Figura 91: Pantalla generación de petición de patrocinio	239
Figura 92: Pantalla de visualización de evento finalizado	240
Figura 93: Pantalla de calificación de evento finalizado	242
Figura 94: Pantalla de visualización de equipos inscriptos	242
Figura 95: Pantalla de visualización del perfil del equipo	243
Figura 96: Pantalla de expulsión del equipo	243
Figura 97: Pantalla de registración del pago del equipo	244
Figura 98: Pantalla de visualización de estadísticas	245
Figura 99: Pantalla del perfil del usuario	246
Figura 100: Pantalla para compartir los diferentes perfiles y eventos	247
Figura 101: Pantalla para contactarse con el administrador	248
Figura 102: Pantalla de registro de resultado de partido	249
Figura 103: Pantalla de visualización de resultados de la jornada.	249
Figura 104: Pantalla de visualización de eventos disponibles	251
Figura 105: Pantalla para establecer la cantidad de publicidad de la aplicación	251
Figura 106: Pantalla de visualización de deportes creados	252
Figura 107: Pantalla de creación de deporte	253
Figura 108: Pantalla de modificación de deporte	254
Figura 109: Pantalla de eliminación de deporte	254
Figura 110: Pantalla de visualización de tipos de eventos creados	255
Figura 111: Pantalla de creación de tipo evento	256
Figura 112: Pantalla de modificación de tipo evento	257
Figura 113: Pantalla de eliminación de tipo evento	257
Figura 114: Pantalla de visualización de tipos de organizadores creados	258
Figura 115: Pantalla de creación de tipo organizador	259
Figura 116: Pantalla de modificación de tipo de organizador	260
Figura 117: Pantalla de eliminación de tipo organizador	261
Figura 118: Pantalla de visualización de roles creados	261
Figura 119: Pantalla de creación de rol	263

Figura 120: Pantalla de modificación de rol	264
Figura 121: Pantalla de eliminación de rol	264
Figura 122: Pantalla de visualización de localidades creadas	265
Figura 123: Pantalla de creación de localidad	265
Figura 124: Pantalla de modificación de localidad	266
Figura 125: Pantalla de eliminación de localidad	267
Figura 126: Pantalla de visualización de departamentos creados	267
Figura 127: Pantalla creación de departamento	268
Figura 128: Pantalla de modificación de departamento	269
Figura 129: Pantalla de eliminación de departamento	270
Figura 130: Pantalla de visualización de provincias creadas	270
Figura 131: Pantalla de creación de provincia	271
Figura 132: Pantalla de modificación de provincia	272
Figura 133: Pantalla de eliminación de provincia	273
Figura 134: Pantalla de visualización de países creados	273
Figura 135: Pantalla de creación de país	274
Figura 136: Pantalla de modificación de país	274
Figura 137: Pantalla de eliminación de país	275
Figura 138: Pantalla de visualización de canchas creadas	275
Figura 139: Pantalla de creación de canchas	276
Figura 140: Pantalla de modificación de cancha	277
Figura 141: Pantalla de eliminación de cancha	278
Figura 142: Pantalla para cambiar email	278
Figura 143: Pantalla para cambiar contraseña	279
Figura 144: Pantalla de visualización de árbitros	280
Figura 145: Pantalla de Creación de Jornadas	281
Figura 146: Pantalla de Creación de Partido	282
Figura 147: Pantalla de Visualización de peticiones de patrocinio como patrocinador	282
Figura 148: Pantalla de Visualización de Eventos Patrocinados Como Patrocinador	283
Figura 149: Pantalla de Visualización de Equipos como deportista / club	284
Figura 150: Pantalla de alta y modificación de equipos	284
Figura 151: Pantalla de visualización de la integración con Mercado Pago	285
Figura 152: Ventana de registro de pagos de equipos en un evento	285
Figura 153: Botón de feedback en categoría de evento	286
Figura 154: Ventana de feedback	287
Figura 155: Pantalla de visualización de las categorías.	287
Figura 156: Pantalla de modificación de las categorías.	288
Figura 157: Creación de deporte favorito	288
Figura 158: Pantalla que nos muestra como editar el perfil desde la perspectiva del deportista.	289
Figura 159: Pantalla que nos muestra como editar el perfil desde la perspectiva del organizador.	290



Figura 160: Pantalla que nos muestra como editar el perfil desde la perspectiva del árbitro.	
291	
Figura 161: Pantalla que nos muestra como editar el perfil desde la perspectiva del patrocinador.	291
Figura 162: Agregar rol a usuario propio	291
Figura 163: Quitar rol a usuario propio	292
Figura 164: Quitar rol a usuario como administrador	292
Figura 165: Agregar rol a usuario como administrador	292
Figura 166: Pantalla que nos muestra como editar el perfil desde la perspectiva del club.	294
Figura 167: Pantalla de alta y modificación de equipos	294
Figura 168: Apartado de backup en el sidebar del administrador	294
Figura 169: Botón para realizar un backup	294
Figura 170: Botón para restaurar un backup	295
Figura 171: Imagen que nos muestra la ubicación de guardado del backup	295
Figura 172: Ventana de eliminación de equipos del evento	295
Figura 173: Visualización de eventos patrocinados actualmente	296
Figura 174: Visualización de mejores equipos	296
Figura 175: Visualización de eventos pendientes de respuesta de patrocinio.	297
Figura 176: Inscribirse a un equipo como deportista	297
Figura 177: Inscribirse a un equipo como deportista	298
Figura 178: Vincular red social	299
Figura 179: Compartir equipo con una red social	299
Figura 180: Diagrama de transición de estados de la inscripción	300
Figura 181: Diagrama de transición de estados del evento.	301
Figura 182: Diagrama de transición de estados de las canchas	302
Figura 183: Diagrama de transición de estados de los deportistas en el evento	303
Figura 184: Diagrama de transición de estados de los patrocinadores en el evento	304
Figura 185: Logo de Angular	315
Figura 186: Logo de Redux	316
Figura 187: Logo Bootstrap	316
Figura 188: Logo de nodeJS	317
Figura 189: Logo de express	317
Figura 190: Logo de Kubernetes	318
Figura 191: Logo de Docker	318
Figura 192: Logo de mongoDB	319
Figura 193 Logo de redis	320
Figura 194: Logo de mongoose	320
Figura 195: Logo de Cloudinary	320
Figura 196: Logo de vs code	321
Figura 197: Logo de WebStorm	321
Figura 198: Logo de JMeter	322
Figura 199: Logo de Jasmine	322

Figura 200: Estructura del equipo de trabajo	330
Figura 201: Logo de WhatsApp	331
Figura 202: Logo de discord	332
Figura 203: Logo de Google drive	332
Figura 204: Logo de moodle	333
Figura 205: Logo de Trello	333
Figura 206: Logo de git	333
Figura 207: Formato del versionado	333
Figura 208: Diagrama de recursos del Scrum Master	339
Figura 209: Diagrama de recursos del desarrollador BackEnd 1	340
Figura 210: Diagrama de recursos del desarrollador FrontEnd 1	340
Figura 211: Diagrama de recursos del desarrollador BackEnd 2	341
Figura 212: Diagrama de recursos del desarrollador FrontEnd 2	341
Figura 213: Diagrama de recursos del analista 1	342
Figura 214: Diagrama de recursos del analista 2	342
Figura 215: Diagrama de recursos del analista 3	343
Figura 216: Diagrama de recursos del diseñador 1	343
Figura 217: Diagrama de recursos del desarrollador diseñador 2	344
Figura 218: Diagrama de recursos del diseñador 3	344
Figura 219: Figura de git	346
Figura 220: Figura de mongoDB	346
Figura 221: Figura de angular	347
Figura 222: Figura de nodeJS	347
Figura 223: Figura de kubernetes	348
Figura 224: Figura de docker	348
Figura 225: Figura de WhatsApp	349
Figura 226: Figura de discord	349
Figura 227: Figura de Google Drive	350
Figura 228: Figura de moodle	350
Figura 229: Figura de trello	351
Figura 230: Costo de guardar imágenes en el repositorio de Google Cloud	361
Figura 231: Costo de utilización de GKE (Google Kubernetes Engine)	361
Figura 232: Gráfica del análisis de los riesgos	363
Figura 233: Gráfica que muestra el porcentaje de cada un de los riegos	364
Figura 234: Logo de GitLab	370
Figura 235: Logo de JMeter	371
Figura 236: Logo de Jasmine	371
Figura 237: Logo Framework Angular	372
Figura 238: Logo de Redux	372
Figura 239: Logo Bootstrap	373
Figura 240: Logo de nodeJS	374
Figura 241: Logo de express	374

Figura 242: Logo de docker	374
Figura 243: Logo de kubernetes	375
Figura 244: Logo de mongoDB	375
Figura 245: Logo de redis	376
Figura 246: Logo de mongoose	377
Figura 247: Logo de Cloudinary	377
Figura 248: Logo de VS Code	377
Figura 249: Logo de WebStorm	378
Figura 250: Estructura del proyecto FrontEnd	380
Figura 251: Estructura del proyecto BackEnd	381
Figura 252: Pantalla de registrarse	391
Figura 253: Pantalla de registrarse como deportista	392
Figura 254: Pantalla de registrarse	408
Figura 255: Pantalla de registrarse como club	409
Figura 256: Pantalla de selección de ubicación del club	410
Figura 257: Pantalla de registro como patrocinador.	422
Figura 258: Pantalla de registro como organizador.	428
Figura 259: Pantalla de registro como árbitro.	435
Figura 260: Pantalla de inicio de sesión.	446
Figura 261: Pantalla que nos muestra la solapa de cerrar sesión.	454
Figura 262: Pantalla de actualización de email.	460
Figura 263: Pantalla de recuperación de contraseña.	470
Figura 264: Pantalla de recuperación de contraseña.	482
Figura 265: Pantalla de contacto.	488
Figura 266: Documentación de apis en readme	494
Figura 267: Reporte de la cantidad de inscripciones por mes	541
Figura 268: Reporte de los deportes más utilizados por deporte	542
Figura 269: Funcionamiento del cert-manager	549
Figura 270: Funcionamiento del backup 1	551
Figura 271: Funcionamiento del backup 2	552
Figura 272: Imagen que muestra la estructura de la implementación	553
Figura 273 y 274: Imagen que muestra la estructura de la implementación de okteto	553
Figura 275: Imagen que muestra la el resultado obtenido en okteto	558
Figura 276: Imagen que muestra la estructura de la implementación	559
Figura 277: Imagen que muestra la configuración de las variables	559
Implementación con Google Cloud Platform	561
Figura 278: Imagen que muestra la estructura de la implementación	561
Figura 279 y 280: Imagen que muestra la estructura de la implementación de google cloud platform	561
Figura 281: Imagen que muestra la el resultado obtenido en okteto	567

## Indice de anexos

ANEXO 1: Diagrama de tiempo de Sportinev Gantt	314
ANEXO 2: Diagrama de casos de uso	304
ANEXO 3: Diagrama de clases	304
ANEXO 4: Manual de usuario de Sportinev	534



# Trabajo Práctico 1

Desarrollo de un sistema de información real.

## Resumen.

Esta idea surgió debido a que los métodos actuales para gestionar los eventos deportivos en Mendoza, en especial eventos deportivos de tenis gestionados por la Federación Mendocina de Tenis son analógicos. Esto genera una dificultad importante a la hora de tratar la información y produce como resultado una ineficiencia que impacta en el servicio dado. Pensando en esto también nos dimos cuenta del problema de la publicidad, puesto que la organización por sí sola se tiene que hacer cargo de esta, con nuestro sistema pretendemos no solo llevar la publicidad directamente a los deportistas, sino que también llamar la atención de los patrocinadores.

Sportinev es un sistema de gestión de eventos deportivos cuyo objetivo es desarrollar un sistema de información que permita a los organizadores crear eventos y llevar un registro de ellos, conectándose con los deportistas y patrocinadores interesados. El impacto consiste en generar una red de organizaciones que posibiliten a los deportistas del sistema tener una gran variedad de opciones, generando así mayor publicidad para los eventos disputados, esto quiere decir que las organizaciones podrán exponer su información hacia los clientes para que estos puedan verse interesados.

## Palabras Clave.

Promoción deportiva, eventos deportivos, incentivo a la actividad física, diversión deportiva, eventos organizados.

## Conclusiones.

Respecto a los logros durante el año de desarrollo de proyecto desde el punto de vista funcional el resultado final es positivo. Según se observó en los eventos deportivos, se carecía de una organización formal en la provincia de Mendoza, además de carecer de una buena difusión a través de medios digitales. Los organizadores necesitaban una metodología eficiente que les permitiera administrar y organizar sus eventos tanto a nivel de horarios como a nivel de inscripciones.

Paralelamente había una fuerte necesidad de parte de los deportistas de tener una herramienta para encontrar y poder inscribirse a eventos según sus preferencias y necesidades.

La parametrización e implementación tecnológica permitió que el sistema se adapte a cualquier deporte ya sea por equipos o en solitario.

Además, se contempló e integró los roles claves en un evento deportivo como patrocinadores y árbitros. Los patrocinadores pueden patrocinar eventos directamente desde la plataforma y ver los perfiles de los organizadores para decidir cuál es la mejor opción de patrocinio. Los árbitros tienen sus perfiles también y podrán ser escogidos para arbitrar diferentes tipos de eventos deportivos.

Con el desarrollo del proyecto se logra tener un solo lugar donde poder satisfacer todas estas necesidades con todas las comodidades que permite hoy la digitalización, con una respuesta confiable y además, integrando el pago en la plataforma permitiendo que los actores puedan elegir otras opciones además del efectivo.

En cuanto a las conclusiones desde el punto tecnológico se concluye que las tecnologías utilizadas fueron las adecuadas para la realización del sistema, por ejemplo Angular nos ofreció una arquitectura completa y modular para poder realizar el front end, respecto del back end la utilización de una base datos no relacional como lo es mongoDB fue especialmente agradable ya que nos permitió una gran flexibilidad y escalabilidad de cara al futuro. Además de que todas las tecnologías utilizadas a lo largo del proyecto son muy utilizadas y testeadas en el ámbito profesional en la actualidad.

## Definición de requerimientos.

### Eventols



Figura 1: Logo Eventols

Inscripción al evento de prueba: <https://eventols.com/e/prueba-5>

#### 1) Relevamiento general.

##### 1.1) De la organización.

Eventols es una plataforma de inscripción y gestión de eventos deportivos, particularmente para carreras atléticas.

Esta consiste en ofrecer al usuario organizador poder administrar su evento a través de esta herramienta digital, dándole la oportunidad de obtener la estadística en tiempo real de lo que está sucediendo con su evento.

##### 1.2) Funciones detectadas.

Entre sus funcionalidades se encuentran:

- **Inscripción Online.**  
En este módulo se le ofrece a los deportistas interesados la oportunidad de inscribirse al evento.



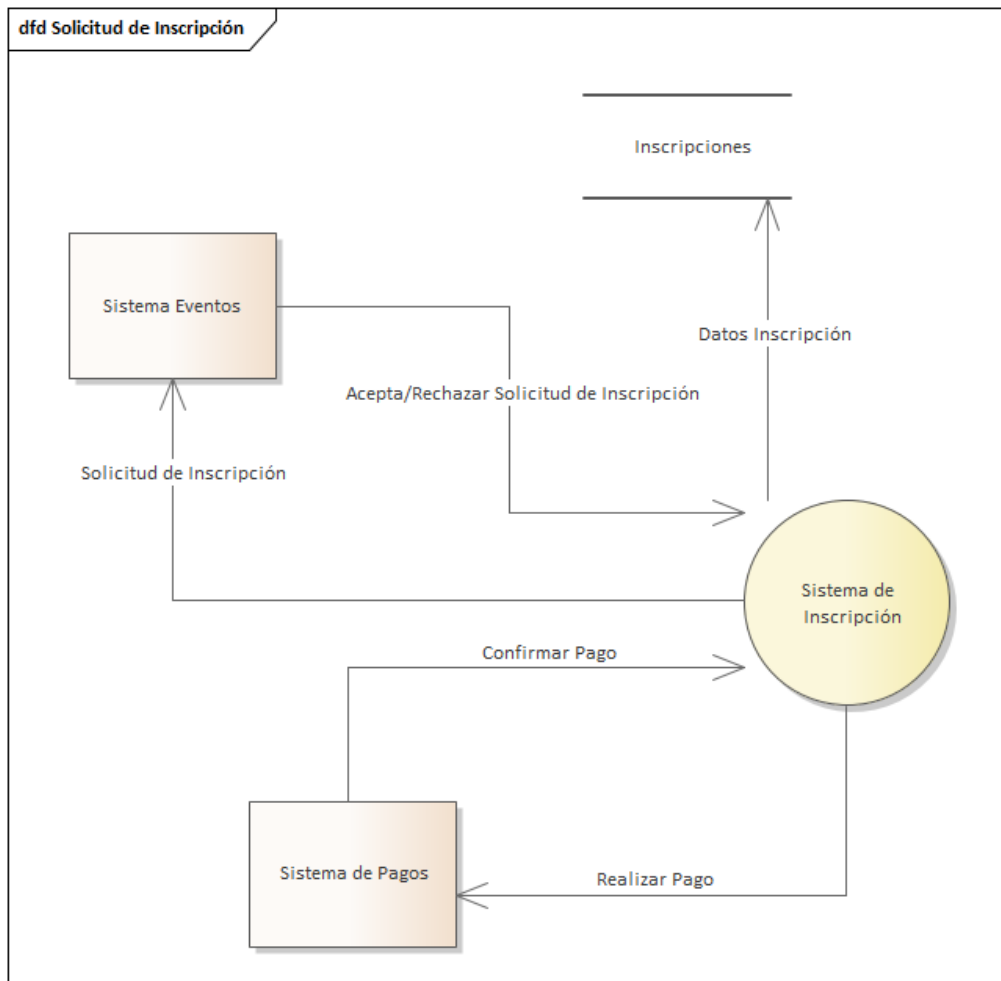


Figura 2: Diagrama de contexto de inscripción online de Eventols

- Pagos Online.
  - En este módulo el deportista podrá realizar el pago para inscribirse al evento.
    - Métodos de pago:
      - Mercado Pago.
      - Transferencia Bancaria.
      - Personalizado (efectivo).

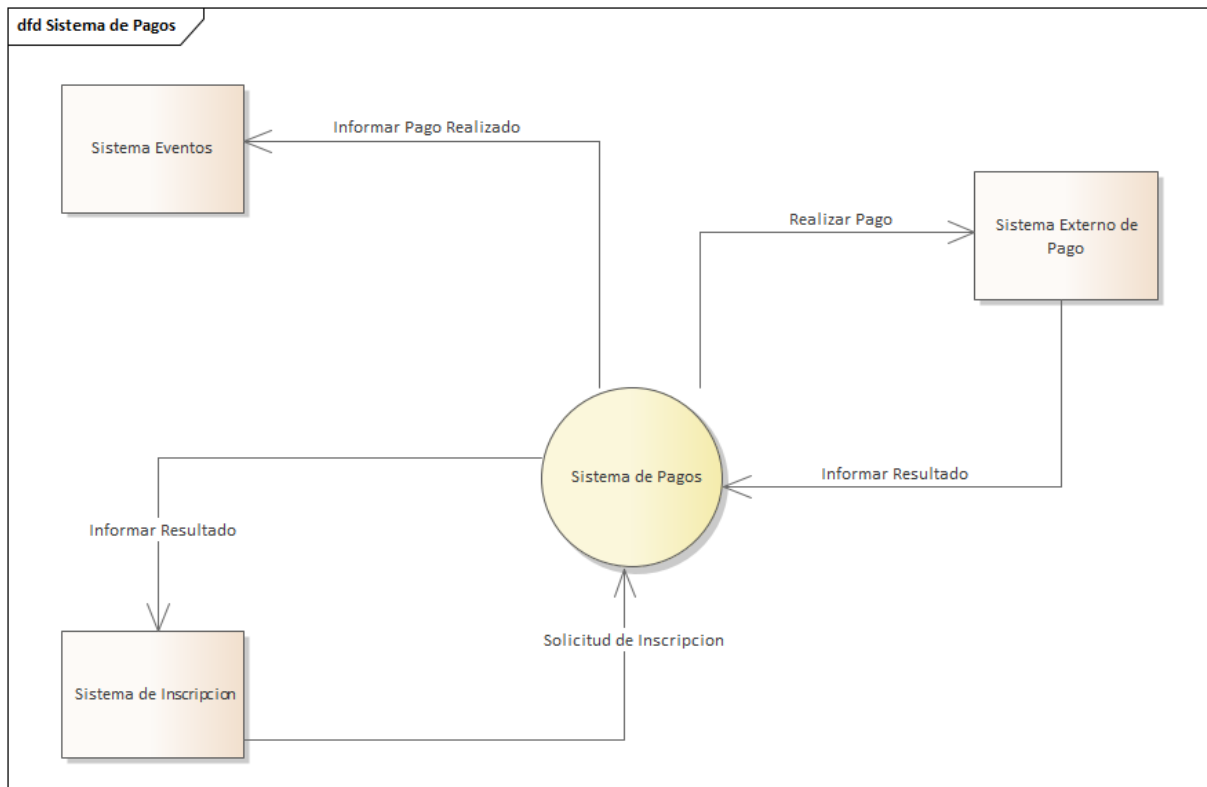


Figura 3: Diagrama de contexto de pagos online de Eventols

- Programar Eventos Deportivos (maratones).
  - Programar Competencias dentro del evento:
- Publicaciones de resultados.
- Administrar categorías del evento.
- Administrar la indumentaria necesaria.
- Administrar usuarios inscriptos.
- Formulario de inscripción customizado.
- Promocionar Evento.
- Resultados de eventos.

### 1.3) Tecnología de información.

Las tecnologías utilizadas en Eventols son:

#### **Sistema operativo.**

- Todos (multiplataforma).

#### **Base de datos.**

- No especifica.

### Lenguaje de programación.

- CSS
- HTML
- JavaScript
- Bootstrap

### Otras herramientas.

- Google Analytics
- Google Tag Manager
- DataTables
- Modernizr
- Select2
- SweetAlert2
- Moment.js 2.18.1
- jQuery 2.2.4
- jQuery UI 1.12.1

### Versión móvil.

- No tiene.

2) Relevamiento detallado y análisis del Sistema.

2.1) Detalle, explicación y documentación detallada de todas las funciones seleccionadas.

### Módulo Eventos.

Este módulo permite al organizador dar de alta un evento, administrarlo, analizarlo en tiempo real, promocionarlo, Clasificación de deportistas.

### Landing Page del evento.

En este apartado podemos ver las estadísticas que definen el comportamiento del evento.

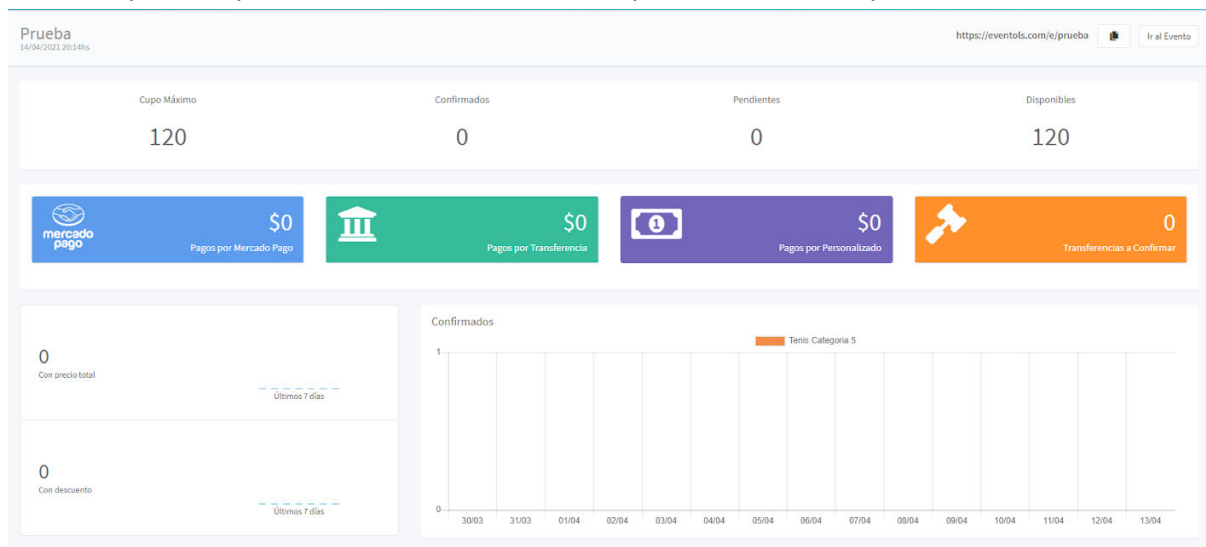


Figura 4: Landing page de Eventols

## Alta de Eventos.

En este módulo debemos caracterizar el evento.

### Datos principales


Nombre   
 Fecha    
 Publicado

Figura 5: Datos principales en Eventos

### Ubicación

Lugar   
 Dirección   
 Ciudad   
 Provincia   
 País   
 Coordenadas  Ingresar coordenadas manualmente



Figura 6: Alta de un evento en Eventos

### Datos adicionales



Imagen Perfil   
  
 Imagen Cover   
  
 Deslinde / Ficha   
 Descripción

Figura 7: Alta de un evento en Eventos

## Modulo Competencias.

En este módulo debemos crear las diferentes competencias que estarán presentes en el evento y asociarlas a una o varias tarifas.

Competición

Nombre   
 Límite inscripciones   
 Visible desde    
 Visible hasta    
 Equipos/Posta   
 Distancia en metros   
 Gratis   
 Solicitar Certificado Médico   
 Activo  Edad mín  Edad max

Tarifas

Precio  \$ Moneda

Figura 8: Configuración de las competencias en Eventols

### Módulo de análisis.

En este módulo podemos acceder a un apartado en tiempo real en el cual podemos ver información relacionada con el comportamiento del/los eventos/s, respecto a los usuarios Registrados/Confirmados/Acreditados filtrados por fecha.

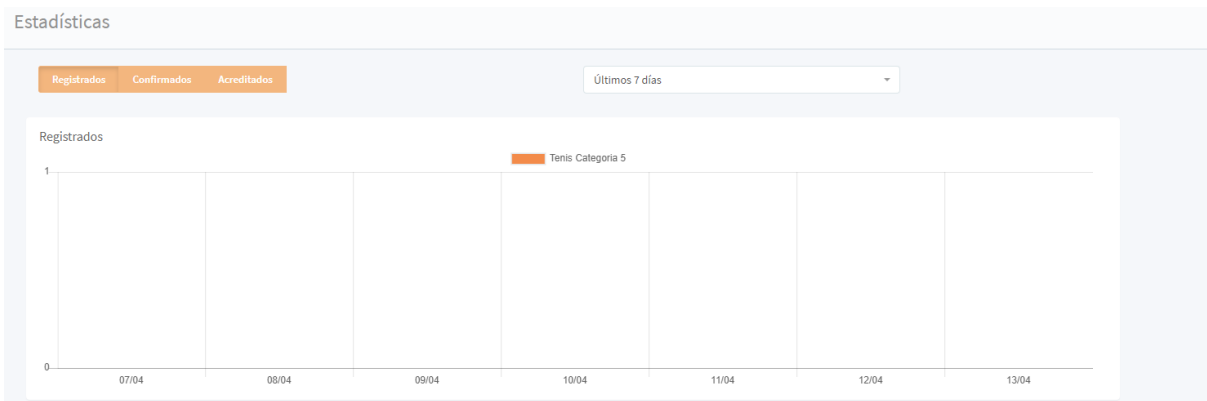


Figura 9: Pantalla principal del módulo de análisis de Eventols

### Módulo de Reportes.

Este módulo permite al organizador generar reportes enfocados a las características del evento.

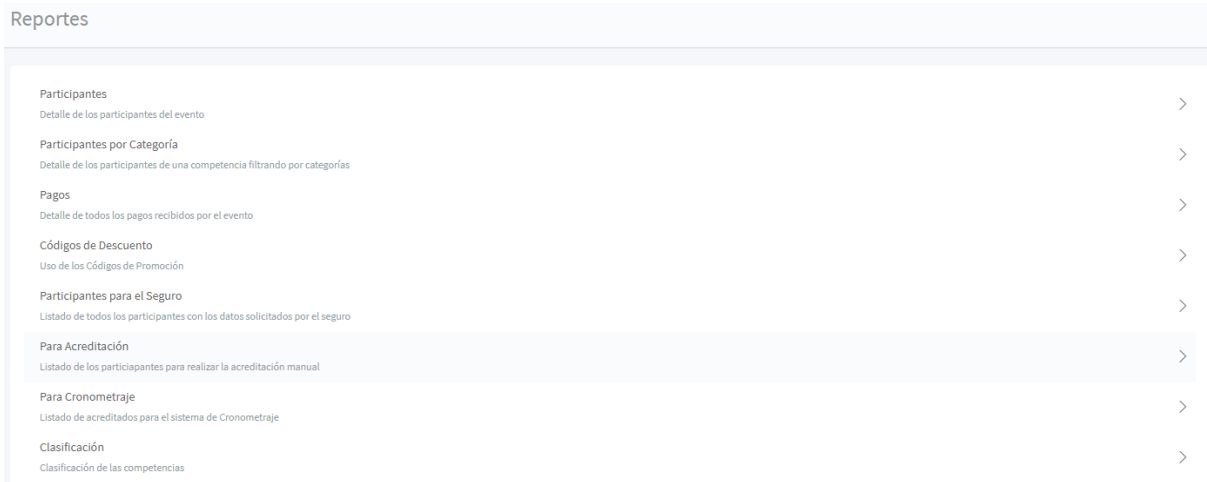


Figura 10: Interfaz del módulo de reportes de Eventols

### Módulo Promoción.

En este módulo el organizador es capaz de diseñar su propia web, la cual representa al evento en cuestión, además de poder customizar las notificaciones del evento hacia los clientes y ofrecer cupones de descuentos en caso de que el evento sea pago.

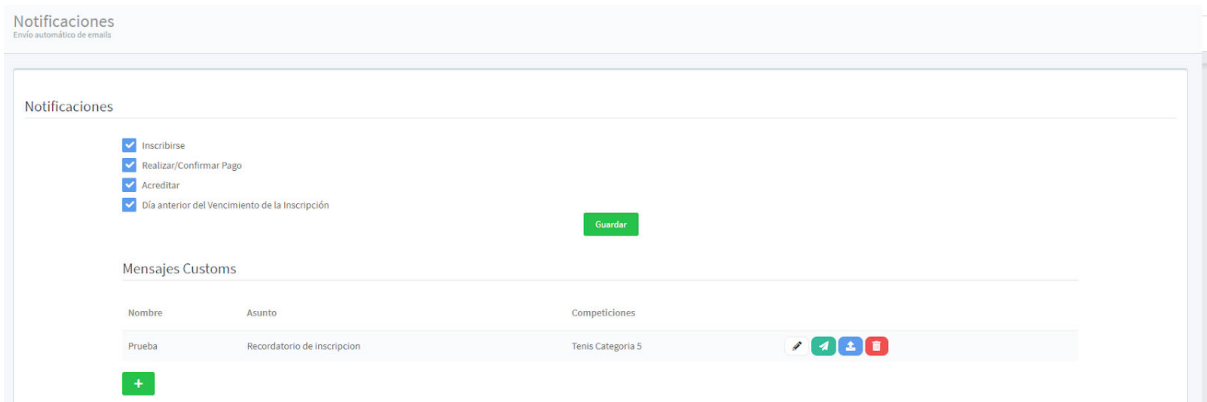





Figura 11: Interfaz del módulo de promociones de Eventols


Cupones de Descuento

Cupones existentes


Código	% de descuento	Límite	Desde	Hasta	
001	10 %		14/04/2021 22:30	15/04/2021 22:30	 


+ 

001

Código:  

% de descuento:

Usos ilimitados 

Fecha desde:  



Fecha hasta:  

Figura 12: Interfaz del módulo de promociones de Eventols

### Módulo Inscripción.

En este módulo el deportista podrá inscribirse al evento y escoger un método de pago, salvo que la competencia no tenga una tarifa asociada, una vez confirmado quedará registrado en la landing page del evento que un deportista se inscribió al evento.

Su reserva expira en 28:11 minutos!

 Categoría de Prueba

Precio: \$500 + \*

Datos de inscripción

Nombre:

Apellido:

Email:

Fecha de Nacimiento:

Sexo:  Hombre  Mujer

País:

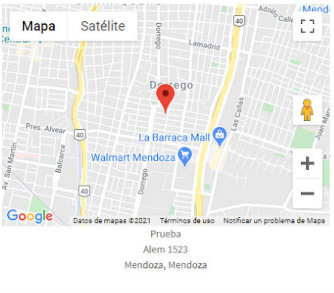
Provincia:

Ciudad:

He leído y acepto las condiciones de uso y las políticas de Eventols. Y acepto el reglamento del evento.

Cuando? miércoles 21 de abril de 2021 a las 00:26hs

Donde?



Ignacio Starkloff

Figura 13: Interfaz del módulo de inscripción de Eventols

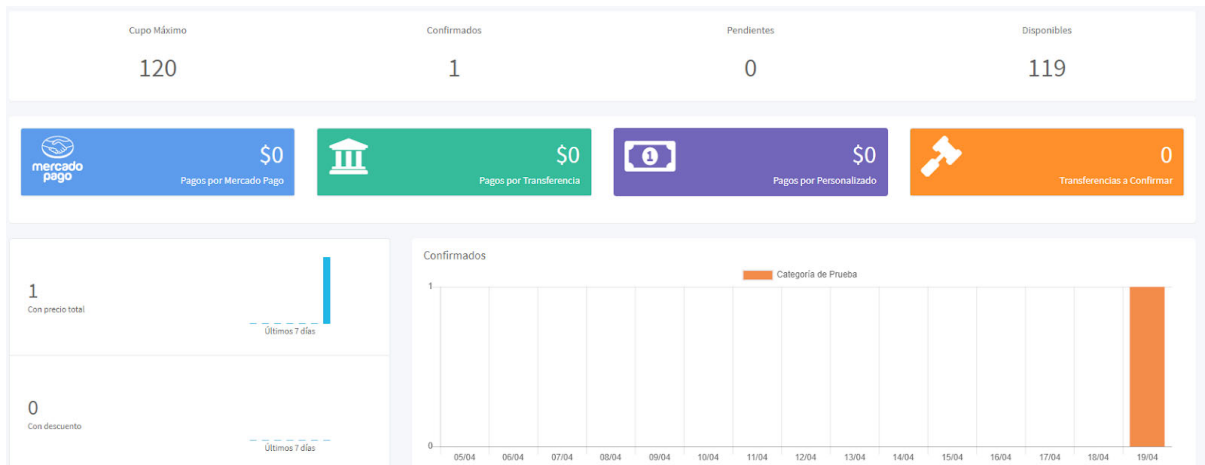


Figura 14: Interfaz del módulo de pagos de Eventols

### Módulo Acreditar Pago.

En este módulo se acreditará la participación de los deportistas inscritos.

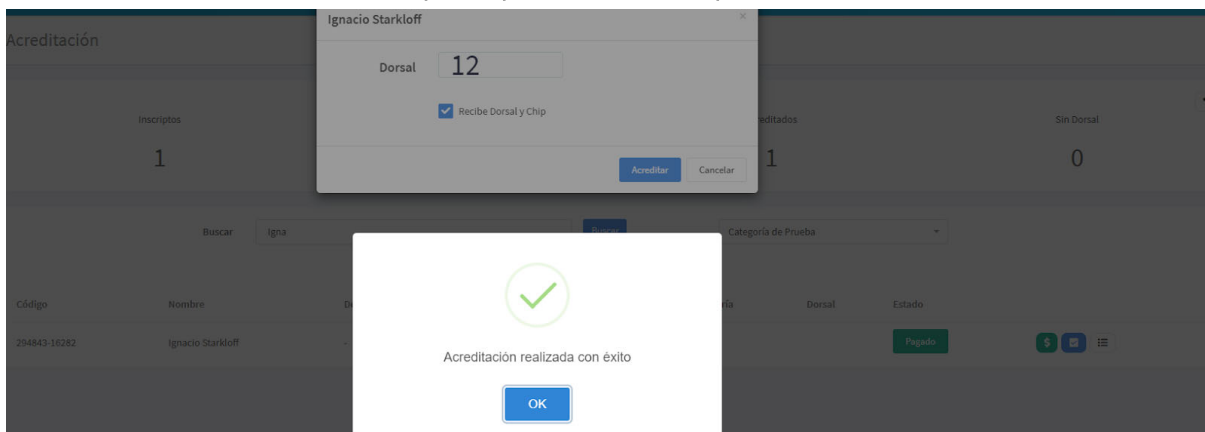


Figura 15: Interfaz del módulo de acreditar de Eventols

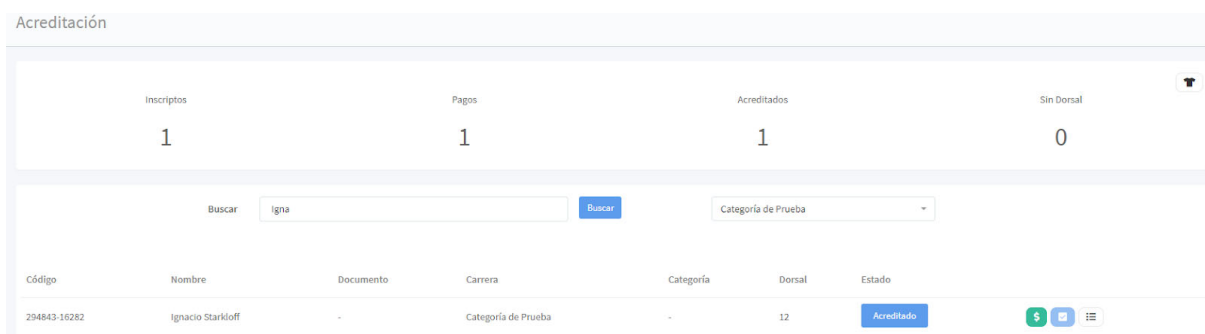




Figura 16: Interfaz del módulo de acreditar de Eventols.

Una vez que el organizador realizó la acreditación del deportista, este podrá ver su estado desde un link que le llegará al email que ingresó:



 Categoría de Prueba	<a href="#">Editar datos</a>
 Ignacio Starkloff	
Precio: \$500.00 + *	
Número: 12	

<b>Al inscribirse</b>	<b>Confirmado</b>
Monto: \$0.00 (+ 0.00)	
Fecha de Pago: 19/04/2021 16:43	
Método de Pago: Bonificado	

Figura 17: Interfaz del módulo de inscripción de Eventos.

## 2.2) Modelo lógico del Sistema actual.

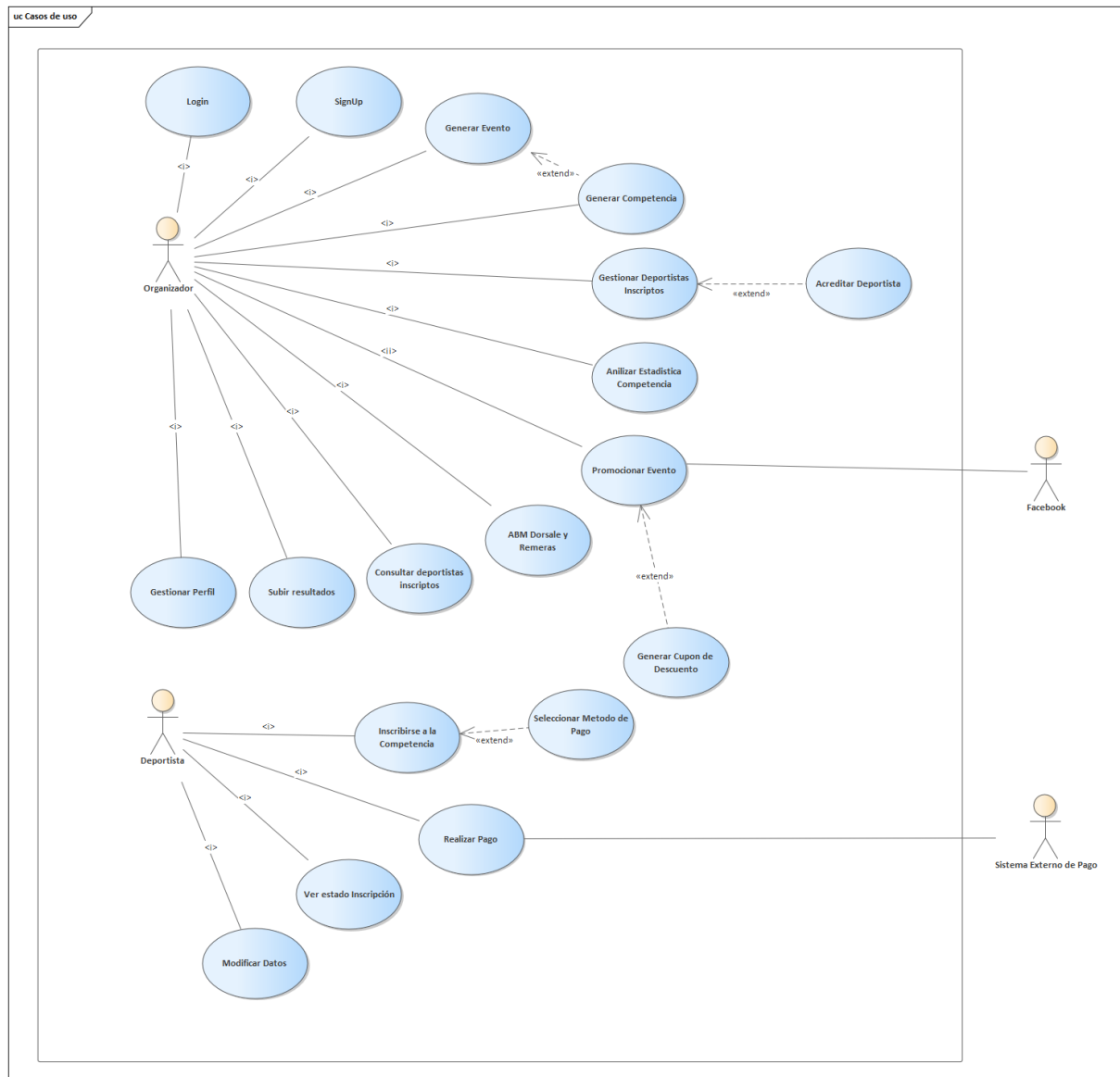


Figura 18: Modelo de casos de uso de Eventols

## 2.3) Problemas y necesidades detectados en las funciones relevadas en detalle y en su entorno organizacional.

### Problemas

- La publicidad del evento depende del organizador.
- Gestión de eventos deportivos exclusivamente para carreras atléticas.
- Dificultad para detallar el lugar del evento.

### Necesidades

- El sistema tiene que dar soporte también a los deportistas, que estos sean parte del sistema y que puedan ver los organizadores registrados en este, junto con sus eventos.
- El sistema podría adecuarse y brindar soporte a los otros deportes, la lógica lo puede permitir, sin embargo, no se decidió hacerlo
- A nuestro parecer, es más cómodo ingresar la ubicación directamente desde Google Maps.

## SportEasy



Figura 19: Logo SportEasy

### 1) Relevamiento general.

#### 1.1) De la organización.

SportEasy es una aplicación para organizar partidos y gestionar equipos de fútbol, rugby, baloncesto o cualquier otro deporte.

En la aplicación se puede administrar los datos del perfil personal, añadiendo datos personales como fotos y deportes que practica; gestionar nuestro equipo deportivo añadiendo escudo colores participantes e información pertinente al mismo, administrar la temporada dando fecha de inicio fecha de fin, y datos de los eventos, añadir eventos ya sean entrenamientos, partidos, eventos especiales, etc.

#### 1.2) Funciones detectadas.

Las funciones detectadas de SportEasy son:

- Calendario, convocatorias, mensajes.
- Sitio web y aplicación móvil.
- Plantel & Perfiles.
- **Mensajes** permite:
  - Iniciar una discusión con quien deseas en el equipo.
  - Cada uno elige su preferencia de notificación (email o notificación push).
  - Enviar noticias de última hora a través de la discusión dedicada al evento.
  - Comparte lo que quieras (fotos, videos, PDF...)
- Calendario permite:
  - Crea el calendario de tu equipo con solamente unos clics.
  - Comparte las informaciones claves (adversario, lugar de encuentro...)
  - Anular o posponer un evento tan simplemente como lo has generado.

- Convocatorias & Presencias permite:
  - Para todos los deportes colectivos, SportEasy te permite elegir a los titulares y los suplentes, y el mejor esquema táctico.
  - Coloca los jugadores disponibles en tu esquema táctico.
  - Utiliza varias alineaciones para un mismo partido.
  - Decides quién puede visualizar la alineación (únicamente los entrenadores por ejemplo).
  - Compártela en las redes sociales.
- Alineaciones de equipo.
- Gestión de las tareas.
- Campeonatos permite:
  - Crear tu campeonato, tus adversarios y los futuros encuentros.
  - Enterarse los resultados de los partidos día tras día.
  - Que todo el equipo siga la evolución de la clasificación.
- Estadísticas y Live Stats permite:
  - Encontrar estadísticas detalladas del equipo y de cada uno de tus jugadores.
  - Entrar en las estadísticas después de cada partido (resultado final, goleadores, asistencias...)
  - Consultar los informes (gráficos y tablas) actualizados automáticamente para seguir el estado de forma del equipo.
  - Hacer un seguimiento individual de tus jugadores.
  - Seguir el partido en directo con el Live Stats.
- **El post-partido** incluye:
  - Informe, notas, votos, comentarios, fotos, videos.
  - Vota por el mejor jugador del partido.
  - Pon nota a tus compañeros de 1 a 10 en función de su rendimiento.
  - Decides quién puede usar estas funcionalidades (por ejemplo, solo los entrenadores).
  - Redacta y comparte el informe del partido.
- Cuotas y colectas.

### 1.3) Tecnología de información.

Las tecnologías utilizadas en SportEasy son:

#### **Sistema operativo**

- Todos (multiplataforma)

#### **Base de datos**

- No especifica

#### **Lenguaje de programación**

- Python
- HTML
- JavaScript
- Bootstrap

### Otras herramientas

- Django
- Nginx
- Google analytics
- Google fonts API
- Modernizr 2.8.3
- Moment.js 2.13.0
- jQuery 2.1.0
- jQuery UI 1.10.4

### Versión móvil

- Cuenta con versiones para [android](#) y [apple](#).

## 2) Relevamiento detallado y análisis del Sistema.

2.1) Detalle, explicación y documentación detallada de todas las funciones seleccionadas.

Módulos funcionales:

**Módulo Perfil:** Es el encargado de administrar los perfiles principales de los usuarios registrados, controla las acciones de los usuarios y las interacciones con los usuarios, equipos, eventos y calendarios. También tiene el deber de administrar los permisos de los usuarios respecto a cada funcionalidad, como equipos, eventos y otros usuarios.

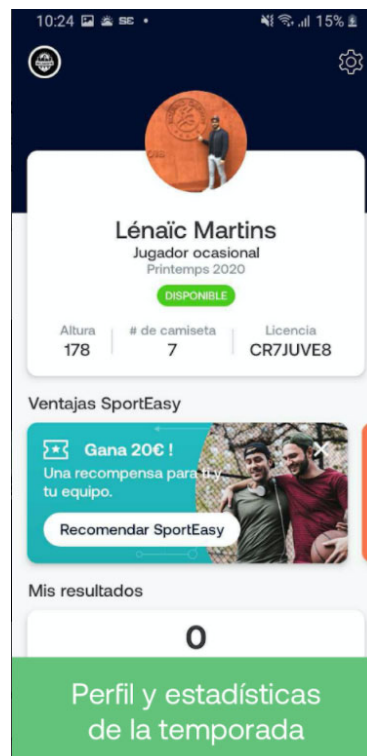


Figura 20: Perfil del deportista en SportEasy

**Módulo Equipos:** Se encarga de gestionar los equipos de los usuarios, pudiendo estos últimos pertenecer a varios equipos o administrar varios, es el encargado de mostrar los equipos y declarar las interacciones entre ellos y con los participantes e integrantes.

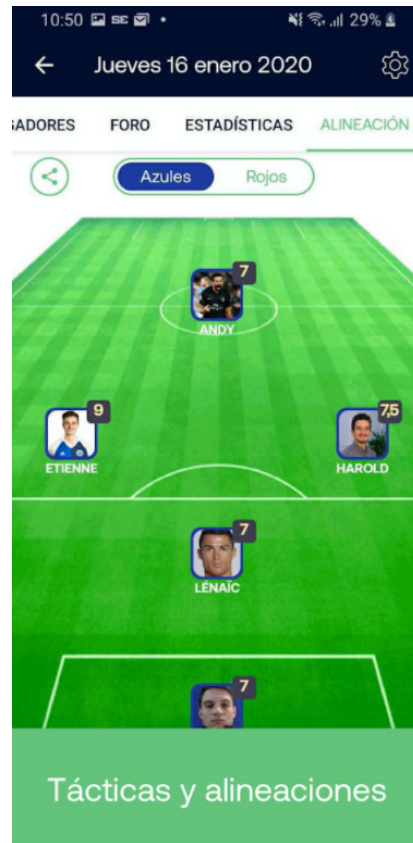


Figura 21: Alineación del equipo en SportEasy

**Modulo Eventos:** Se encarga de administrar los eventos y tipos de eventos en los que se incluyen partidos, enfrentamientos, entrenamientos deportivos, torneos, campeonatos, suspensión de los mismos, retrasos o cualquier tipo de cambios que puedan ocurrir en estos.

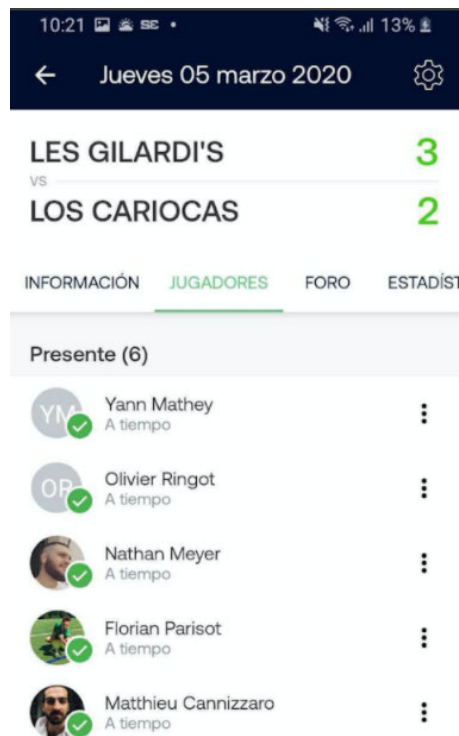


Figura 22: Interfaz de resultados de SportEasy

**Módulo Calendario:** Tiene la responsabilidad de administrar y gestionar todos los eventos de los equipos o usuarios, cuando el usuario los agrega, los guarda y genera reportes y alertas, como recordatorios al usuario, para su asistencia.

**Módulo Estadísticas:** Maneja todas las estadísticas posibles de los equipos, eventos, perfiles, y calendarios agrupándolos por temporadas para brindar una interfaz amigosa representando los datos con diferentes tipos de diagramas.

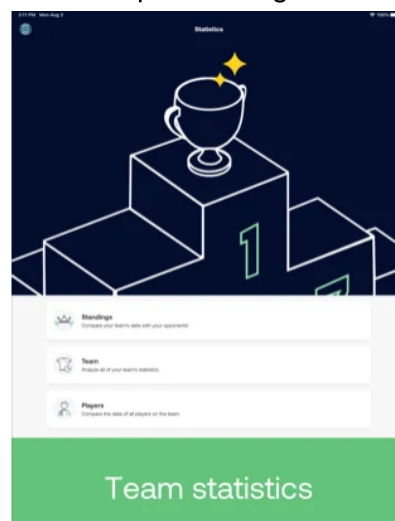


Figura 23: Interfaz del módulo de estadísticas de SportEasy

**Módulo mensajería:** Es el módulo encargado de brindar un servicio de mensajería entre distintos usuarios y usuarios a través de equipos. Puede ser entre dos usuarios o entre varios usuarios simultáneamente utilizando la función de chat grupal, además debe poseer la capacidad de distinguir a los usuarios según el equipo por el que quieren ser identificados y por los permisos concedidos.



Figura 24: Interfaz del módulo de mensajería de SportEasy

**Módulo de Pagos:** Este módulo se encarga de llevar a cabo las transacciones financieras y gestionarlas según su forma de pago dentro de los distintos eventos de la plataforma, asegurando la seguridad del cliente, tratando de evitar fraude o estafas.

## 2.2 Modelo Lógico del sistema actual.

Diagrama de casos de uso:



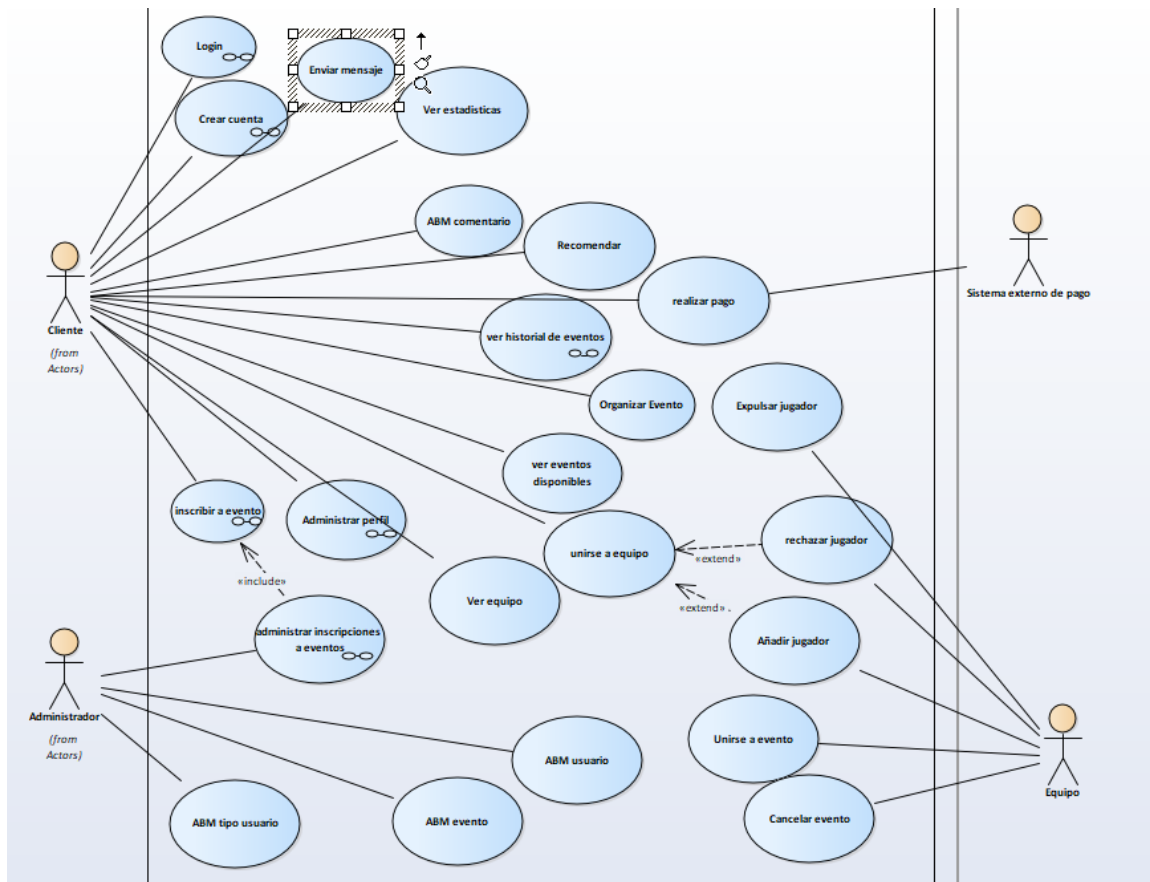


Figura 25: Modelo de casos de uso de SportEasy

## EventBase



Figura 26: Logo eventbase

### 1) Relevamiento general.

#### 1.1) De la organización.

Eventbase es una compañía destinada a la programación de aplicaciones web y móviles, enfocadas en la gestión y la administración de eventos de distinta índole. Una de sus aplicaciones es llamada JOIN IN, la cual administra eventos deportivos.



Figura 27: Logo Join In

La aplicación, si bien permite la gestión de los eventos solamente para los espectadores y los organizadores del evento, es interesante mencionarla, ya que provee varias características que no fueron tenidas en cuenta en la etapa inicial del planteo del sistema, en nuestro proyecto.

Estas características se describen a continuación, con sus funcionalidades.

## 1.2 Tecnología de Información.

### **Sistema operativo**

- Todos (multiplataforma).

### **Base de datos**

- No específica.

### **Lenguaje de programación**

- HTML
- JavaScript
- Bootstrap

### **Otras herramientas**

- Node
- Nginx
- Google analytics
- Google fonts API
- Moment.js
- jQuery
- animate.style

### **Versión móvil**

- No cuenta aún con versión móvil.

## 2) Relevamiento detallado y análisis del Sistema.

2.1) Detalle, explicación y documentación detallada de todas las funciones seleccionadas.

### **Módulo gamificación:**

Impulsa la adopción de aplicaciones y mejora la experiencia de sus espectadores con gamificación y funciones sociales. Nuestro juego anima a los espectadores a ganar puntos y competir por premios completando tareas dentro de la aplicación, como compartir en las redes sociales, encontrar una palabra secreta o visitar una zona de patrocinadores. Utilice nuestros juegos pre configurados o diseñe los suyos propios y muestre los resultados en una tabla de clasificación en todos sus lugares. Con Activity Stream, los espectadores pueden compartir fotos, publicar sobre sus experiencias en eventos y comentar las publicaciones directamente a través de la aplicación.

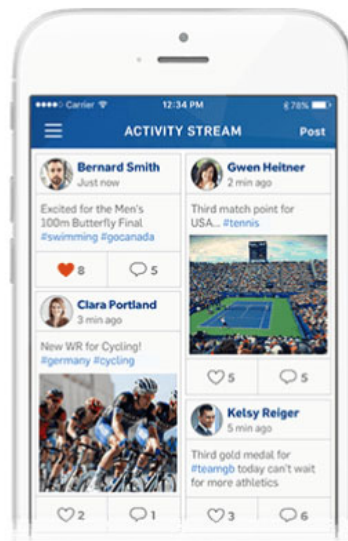


Figura 28: Interfaz del módulo de gamificación de EventBase

### Módulo mensajería localizada:

Los organizadores de eventos pueden usar la tecnología de balizas para ubicar a los asistentes en el interior y enviar mensajes específicos con información sobre patrocinadores cercanos, próximos eventos o partidos e incluso mantener conversaciones por texto con otros usuarios de la aplicación cercanos o específicos.

### Módulo navegación:

Ofrece la posibilidad de navegación GPS para llegar al lugar del evento fácilmente y para representar de una forma interactiva los eventos deportivos a llevarse a cabo en los lugares resaltados de un mapa. Además de poseer un mapa detallado por los organizadores del sitio de cada evento con los puntos de interés, para la comodidad del usuario.

Trabajando con nuestros socios en Heidelberg Mobile, ofrecemos tecnología 3D Deep Map que puede abarcar sitios y edificios de varios niveles. Los usuarios pueden hacer zoom sin problemas y ver un perfil detallado de cada área definida.

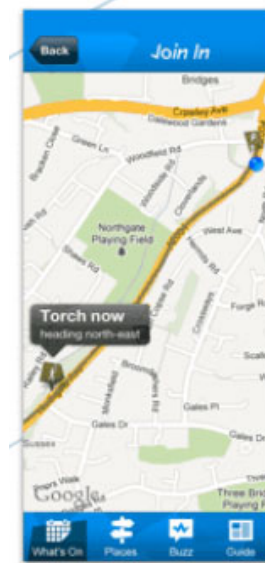


Figura 29: Interfaz del módulo de navegación de EventBase

**Módulo evento:**

Se encarga de administrar los eventos y tipos de eventos en los que se incluyen partidos, enfrentamientos, entrenamientos deportivos, torneos, campeonatos, suspensión de los mismos, retrasos u otros tipos de cambios que puedan surgir en estos.

**Módulo agenda:**

Se encarga de administrar los eventos en la línea de tiempo y trabaja en conjunto con el módulo navegación para localizarlos en el espacio. Además, organiza las agendas de tiempo de los usuarios mostrándole cada actividad incluida en el evento al que este está inscripto.

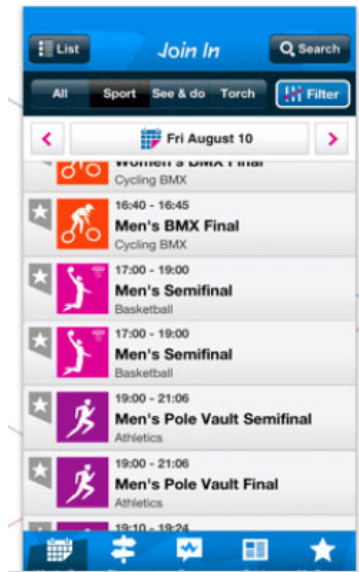


Figura 30: Interfaz del módulo de agenda con los eventos próximos de EventBase

2.2) Modelo lógico del Sistema actual.

**Diagrama de casos de uso.**

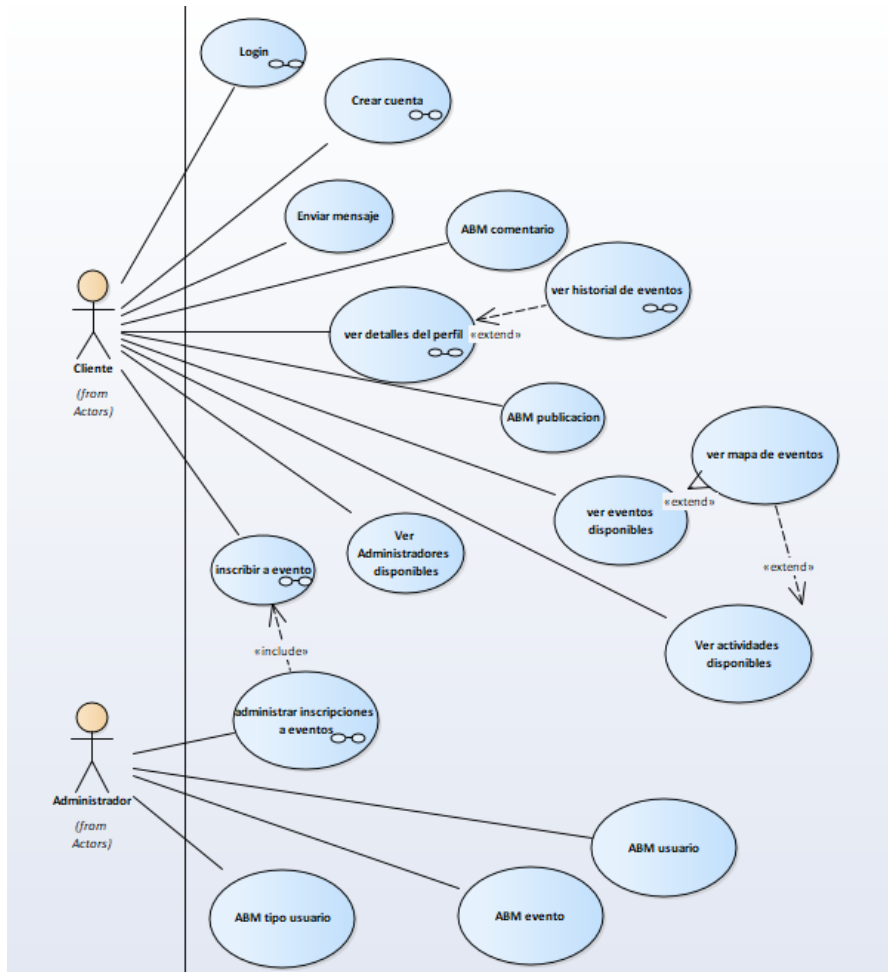


Figura 31: Modelo de casos de uso de EventBase

## Diagrama de contexto.

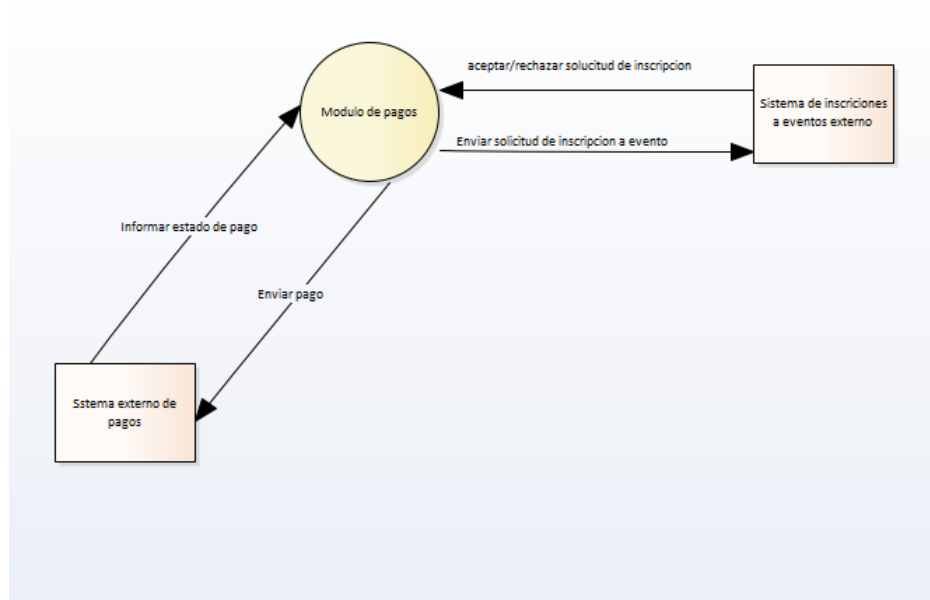


Figura 32: Diagrama de contexto para el módulo de pagos de EventBase

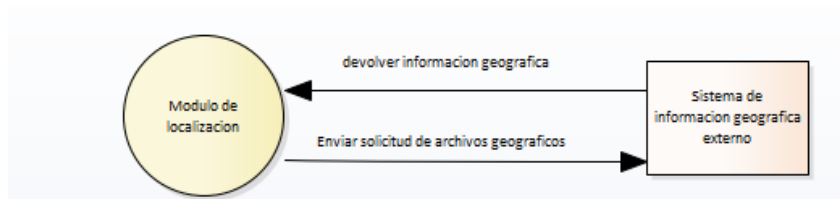


Figura 33: Diagrama de contexto para el módulo de localización de EventBase

2.3) Problemas y necesidades detectados en las funciones relevadas en detalle y en su entorno organizacional.

### Problemas

- La interfaz especialmente en pantallas pequeñas no es la más adecuada.
- La interfaz es poco intuitiva.
- El chat de la aplicación es bastante limitado en cuanto a funcionalidad.
- La aplicación no gestiona bien las fechas y las colisiones de eventos.
- Los usuarios normales tienen funciones muy limitadas.
- La aplicación no gestiona bien los pagos de las inscripciones a eventos dejándolo en manos del usuario y el administrador con quien debe comunicarse cara a cara.

### Necesidades

- Crear una interfaz totalmente responsive.
- Realizar una interfaz intuitiva y amigable que oriente al usuario
- Las estadísticas de los equipos podrían ser más detalladas.
- Gestionar los eventos y las colisiones entre cada evento y partido para evitar superposiciones sugiriendo fechas ininterrumpidas.
- Dar a los usuarios más funcionalidades como elegir su tipo de usuario o crear eventos.
- Generar diferentes funciones de chat para mensajería grupal y mensajería según contactos.
- Gestionar los pagos de las inscripciones a eventos virtualmente

## Playinga



Figura 34: Logo Playinga

### 1) Relevamiento general

#### 1.1) De la organización

Playinga es una aplicación india para gestionar eventos deportivos, la cual permite organizar torneos, gestionar equipos, anotar partidos y realizar un seguimiento del rendimiento de tu equipo en cada partido jugado. Su principal característica es la posibilidad de utilizarla gratuitamente.

PLAYINGA se centra en crear una plataforma deportiva digital integral, que brinda la capacidad de mantener su perfil deportivo y construir una red deportiva para varios deportes.

#### 1.2) Funciones detectadas e interfaces

Las funciones detectadas en playinga son:

- Organización de torneos dinámica.
- Torneos Y Ligas como medio para obtener ganancias, cobrando las tarifas de registro.
- Registro sencillo en equipos, deportes y eventos.
- Generación automática de calendario para torneos.
- Técnica de eliminación en eventos Round Robin Fixture.
- Actualización automática de la progresión y la clasificación de los participantes del torneo.



- Formularios de cobro y pagos.
- Gestión de facturas de torneos.
- Posibilidad de exportación de la aplicación como componente web.
- Creación de álbumes y almacenamiento de recuerdos fotográficos.
- Clasificaciones personalizadas: Puede agregar puntos de posición específicos de deportes personalizados como Tasa de carrera neta, Puntos de juego limpio, etc. e ingresar valores en cada partido.
- Actualizaciones de los eventos por correo electrónico
- Estadísticas y reportes de la aplicación.
- Roles y permisos para cada usuario y tipo de usuario.
- Multideporte - multidisciplinario.

### 1.3) Tecnología de Información

#### **Sistema operativo**

- Todos (multiplataforma).

#### **Base de datos**

- No especifica.

#### **Lenguaje de programación**

- CSS
- HTML
- JavaScript
- Bootstrap

#### **Otras herramientas**

- Google Analytics
- Google Tag Manager
- webpack
- Chart.js
- zone.js
- node.js
- zoho
- DataTables
- Modernizr
- Select2
- SweetAlert2

#### **Versión móvil**

- No tiene.

## 2) Relevamiento detallado y análisis del Sistema.

2.1) Detalle, explicación y documentación detallada de todas las funciones seleccionadas.

**Módulo de creación de equipos:** Este módulo nos permite crear nuestro propio equipo para que luego nuestros amigos se puedan unir. Como información básica nos pide el nombre y el nombre corto.

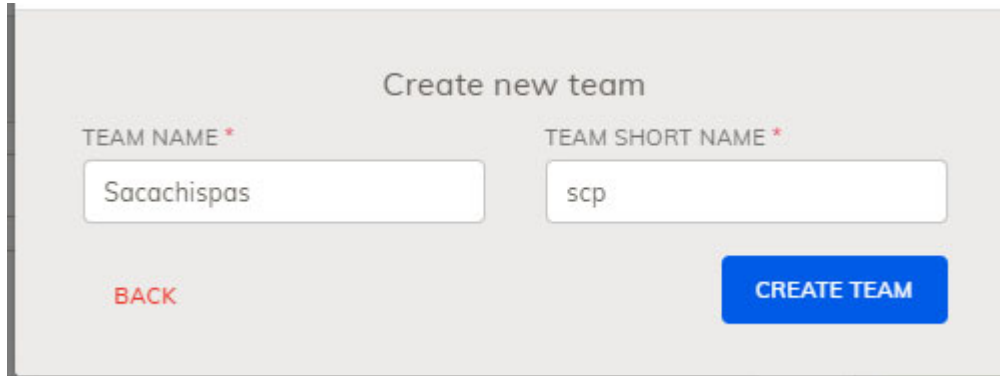


Figure 35 shows a web form titled "Create new team". The form contains two input fields: "TEAM NAME \*" with the text "Sacachispas" and "TEAM SHORT NAME \*" with the text "scp". Below the input fields, there is a red "BACK" link on the left and a blue "CREATE TEAM" button on the right.

Figura 35: Pantalla para la creación del equipo

**Módulo de creación de partidos aislados:** Este pequeño módulo nos permite la creación de un partido "amistoso" en el cual 2 o más equipos se juntan en un lugar para organizar un partido, en una fecha y hora determinada.

### Schedule Match ✕

**MATCH NAME \***

**DATE \*** **TIME \***

**SPORT \***

**CHOOSE YOUR TEAM \*** **CHOOSE OPPONENT TEAM \***

**LOCATION (CHOOSE FROM DROPDOWN) \***

Show Map

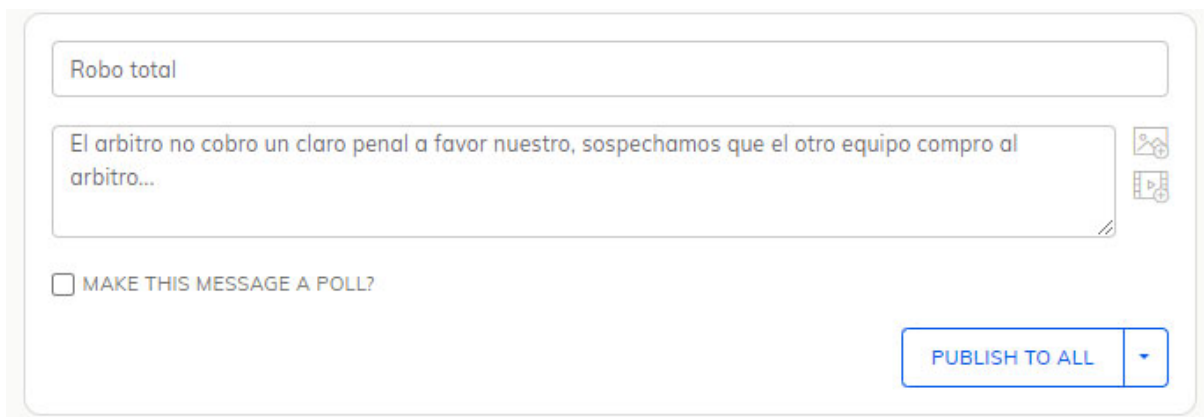
SCHEDULE MATCH

Figura 36: Pantalla para la organización partido



Figura 37: Pantalla del partido de playinga

**Módulo de comentarios:** El módulo de comentarios permite a los miembros de los equipos hacer comentarios del partido, con la posibilidad de insertar fotos y videos del partido realizado.



Robo total

El arbitro no cobro un claro penal a favor nuestro, sospechamos que el otro equipo compro al arbitro...

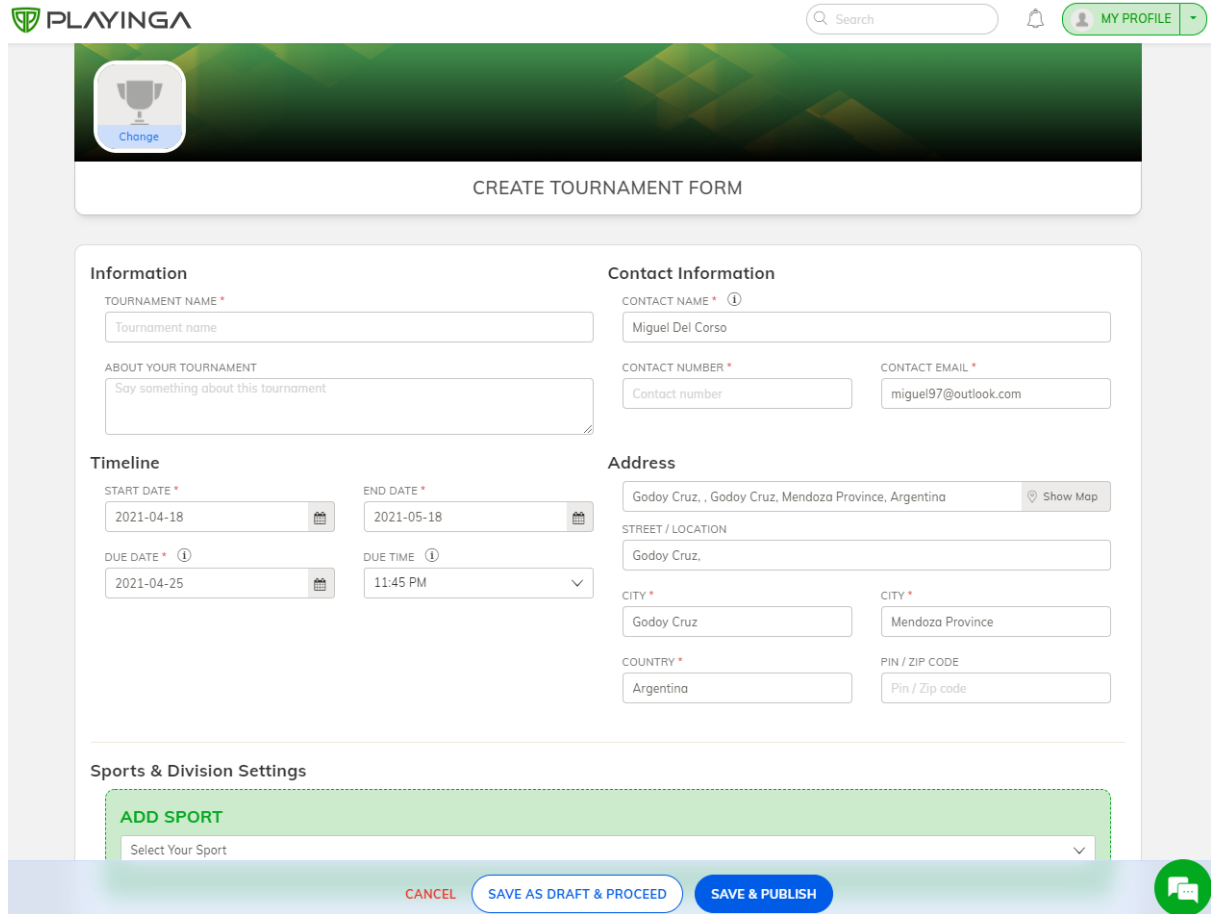
MAKE THIS MESSAGE A POLL?

PUBLISH TO ALL

Figura 38: Interfaz de comentarios de playinga

**Módulo de creación de torneos:** El módulo de creación de torneos nos permite crear un torneo para un deporte en particular, elegir el tipo de torneo (liga o copa), el número máximo de equipos, el nombre de la división en caso de que sea una liga de una forma simple y rápida.

Una característica interesante del módulo es que nos permite elegir el sexo (masculino o femenino) que necesitan los participantes para participar, además de un límite máximo de edad. Estas limitaciones están pensadas para poder hacer las cosas más equilibradas y competitivas.



**PLAYINGA** Search MY PROFILE

**CREATE TOURNAMENT FORM**

**Information**

TOURNAMENT NAME \*  
Tournament name

ABOUT YOUR TOURNAMENT  
Say something about this tournament

**Timeline**

START DATE \* 2021-04-18  
END DATE \* 2021-05-18  
DUE DATE \* 2021-04-25  
DUE TIME 11:45 PM

**Contact Information**

CONTACT NAME \* Miguel Del Corso  
CONTACT NUMBER \*  
CONTACT EMAIL \* miguel97@outlook.com

**Address**

Godoy Cruz., Godoy Cruz, Mendoza Province, Argentina  
STREET / LOCATION: Godoy Cruz.  
CITY \* Godoy Cruz, Mendoza Province  
COUNTRY \* Argentina  
PIN / ZIP CODE: Pin / Zip code

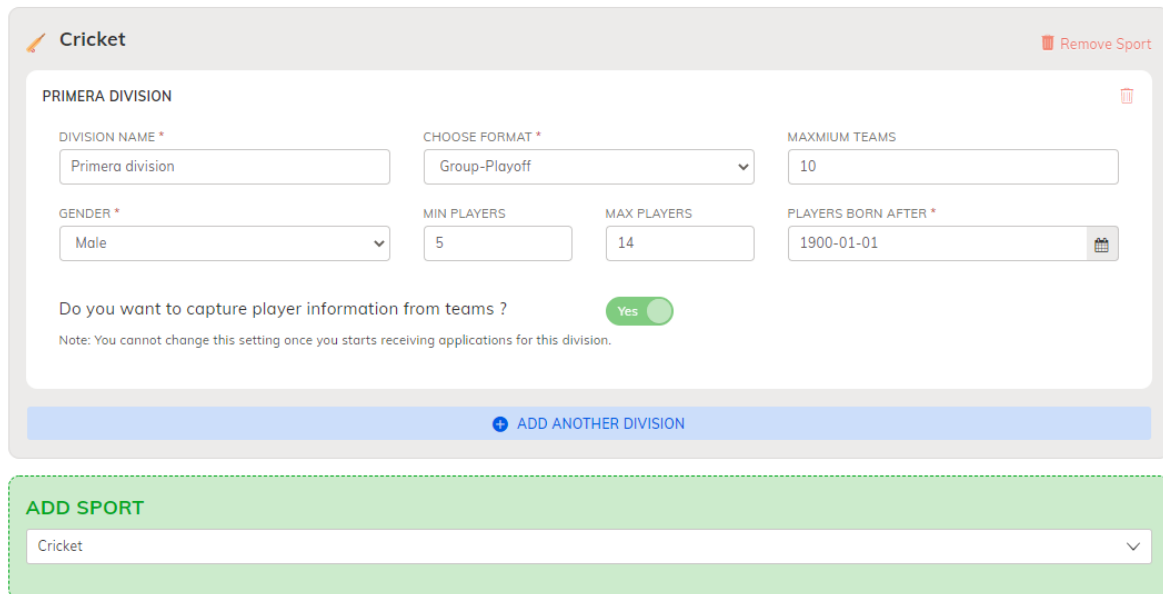
**Sports & Division Settings**

ADD SPORT  
Select Your Sport

CANCEL SAVE AS DRAFT & PROCEED SAVE & PUBLISH

Figura 39: Interfaz de creación de un torneo en playinga

Sports & Division Settings



Cricket Remove Sport

**PRIMERA DIVISION**

DIVISION NAME \* Primera division  
CHOOSE FORMAT \* Group-Playoff  
MAXIMUM TEAMS 10  
GENDER \* Male  
MIN PLAYERS 5  
MAX PLAYERS 14  
PLAYERS BORN AFTER \* 1900-01-01

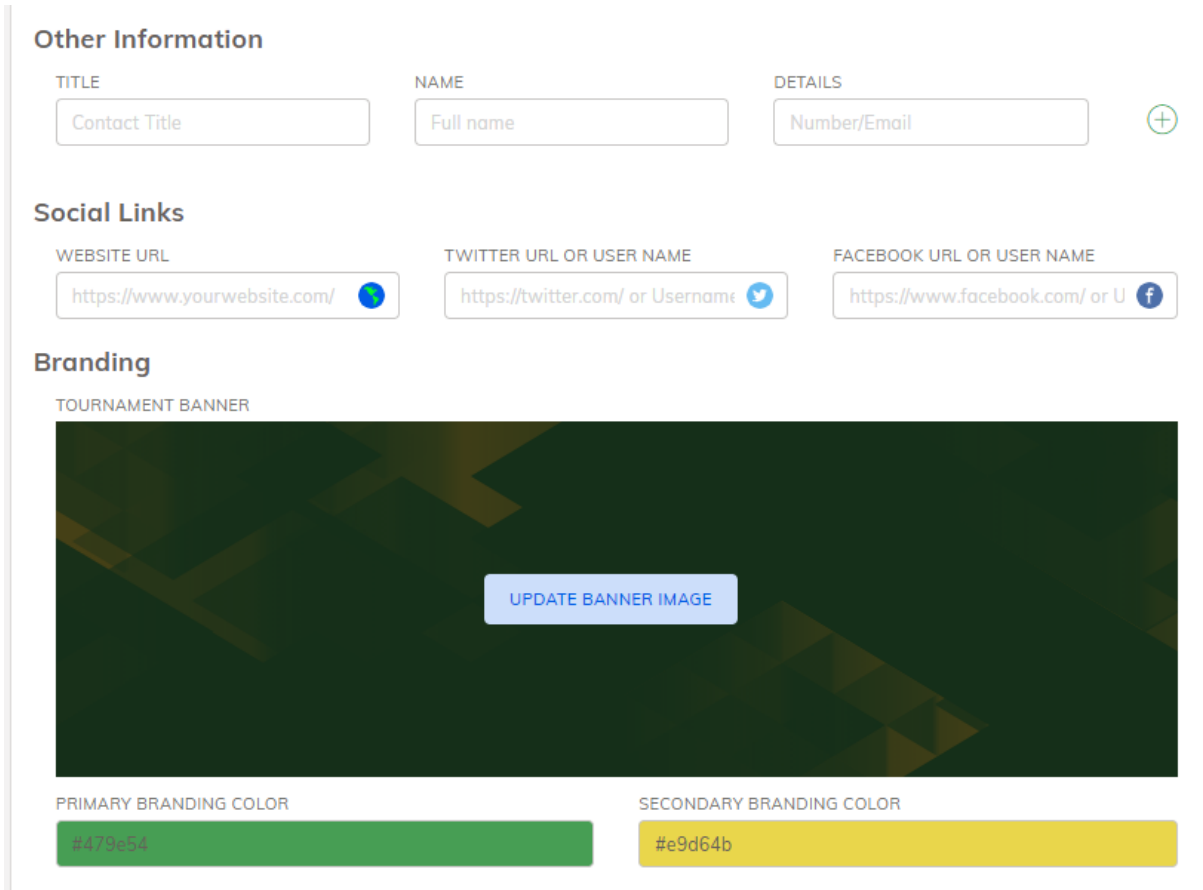
Do you want to capture player information from teams ? Yes

Note: You cannot change this setting once you starts receiving applications for this division.

ADD ANOTHER DIVISION

ADD SPORT  
Cricket

Figura 40: Interfaz de creación de un torneo en playinga



The screenshot displays a web form for creating a tournament. It is divided into three main sections:

- Other Information:** Contains three input fields: 'TITLE' (with placeholder 'Contact Title'), 'NAME' (with placeholder 'Full name'), and 'DETAILS' (with placeholder 'Number/Email'). A plus sign icon is visible to the right of the 'DETAILS' field.
- Social Links:** Contains three input fields for social media links: 'WEBSITE URL' (placeholder: 'https://www.yourwebsite.com/'), 'TWITTER URL OR USER NAME' (placeholder: 'https://twitter.com/ or Username'), and 'FACEBOOK URL OR USER NAME' (placeholder: 'https://www.facebook.com/ or U'). Each field has a corresponding social media icon.
- Branding:** Features a large image area for the 'TOURNAMENT BANNER' with a blue 'UPDATE BANNER IMAGE' button. Below the banner are two color selection boxes: 'PRIMARY BRANDING COLOR' (hex code #479e54) and 'SECONDARY BRANDING COLOR' (hex code #e9d64b).

Figura 41: Interfaz para poder personalizar el “branding” del equipo

Otro aspecto interesante de la creación de torneos es que nos permite hacer un “branding” del torneo, pudiendo elegir los colores del torneo, una imagen que lo representa, y enlaces de contacto útiles.

**Módulo de pagos:** El módulo de pagos permite cobrar tarifas de inscripción para unirse a un torneo. Playinga le permite al organizador del torneo saber al instante qué pagos y jugadores necesitan su atención. Este módulo utiliza Stripe que ofrece un mecanismo más conveniente y seguro de pagar con el procesamiento de pagos en línea.

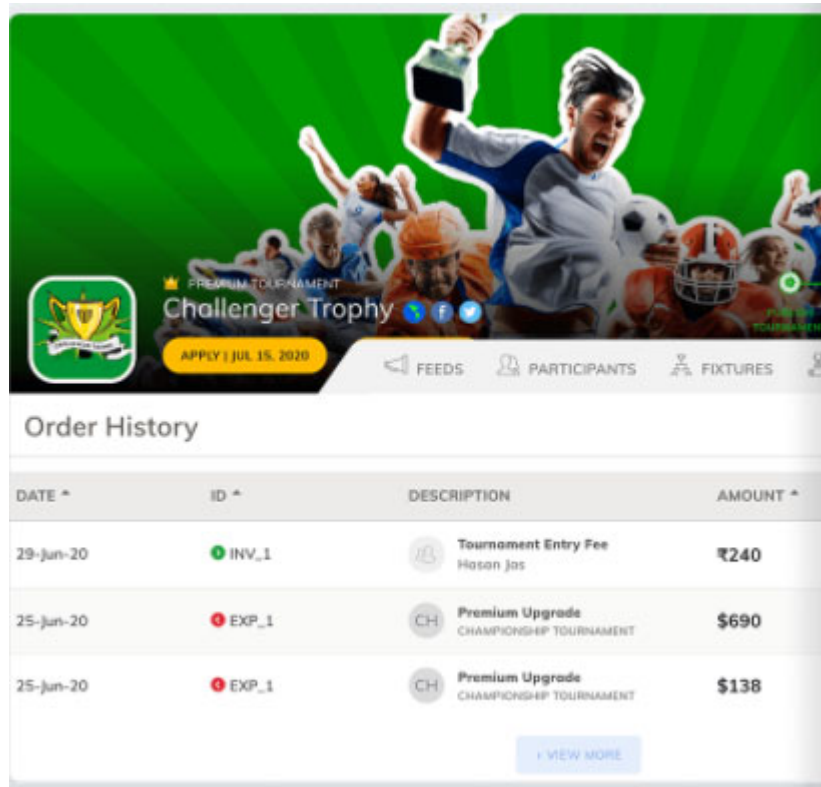



Figura 42: Interfaz del módulo de pagos de playinga

**Módulo de facturación:** El módulo de facturación permite generar una factura lista para presentar a la distancia de un clic, con atributos como la fecha de creación y un detalle de la factura, con su precio correspondiente.



**Challenger Trophy**  
 Moa Creek  
 Otago, NZ  
 9387

✕

Paid  

## INV-108

**BILL TO**

**Jack Stockman / Challenger Trophy**  
 173  
 Berrima  
 New South Wales, AU  
 2577

Date : 30/10/2019  
 Terms : Immediate  
 Due Date : 30-10-2019

---

**ITEMS**

ITEM ID	ITEM DESC	CATEGORY	QTY & UNIT	RATE / UNIT (IN \$)	TAX	AMOUNT
1	Tournament Entry Fee	Enrollment Fee	1 Each	\$ 10.00	vat	\$ 10.00
2	Soccer Mens-Division Fee	Enrollment Fee	1 Each	\$ 50.00	vat	\$ 50.00
3	Badminton Mixed Doubles-Division Fee	Enrollment Fee	4 Each	\$ 40.00	vat	\$ 40.00
4	Womens Volleyball-Division Fee	Enrollment Fee	1 Each	\$ 50.00	vat	\$ 50.00
5	Mens singles-Division Fee	Enrollment Fee	4 Each	\$ 80.00	vat	\$ 80.00
6	T-Shirt	Enrollment Fee	0 Each	\$ 10.00	vat	\$ 0.00

Sub Total	\$230.00
Discount ( - )	\$230.00
Adjustment (+ or -)	\$230.00
<b>Total</b>	<b>\$230.00</b>

Figura 43: Facturación en playinga

**Módulo de retiro de jugador:** El módulo de retiro de jugador permite hacer que un jugador ya no se pueda inscribir a nuevos torneos, ni participar en amistosos. Una vez que el jugador se retiró, cuando ingresamos a su perfil nos muestra sus estadísticas y los torneos que ganó en su etapa como jugador.



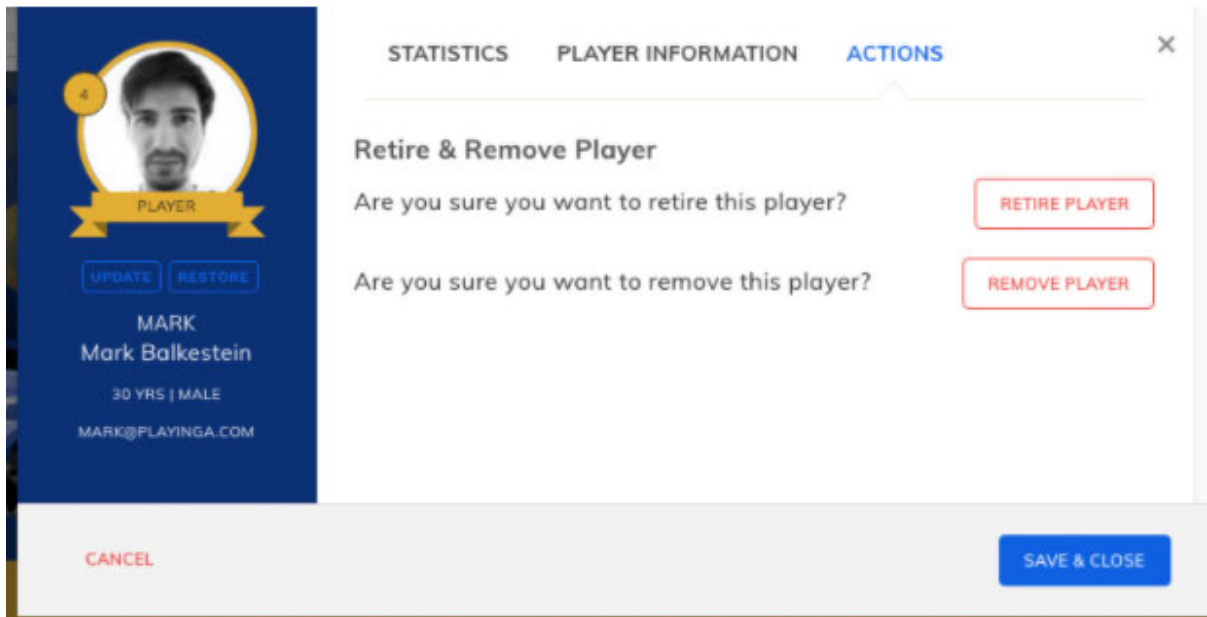


Figura 44: Retiro de un jugador en playinga

2.2) Modelo lógico del Sistema actual.

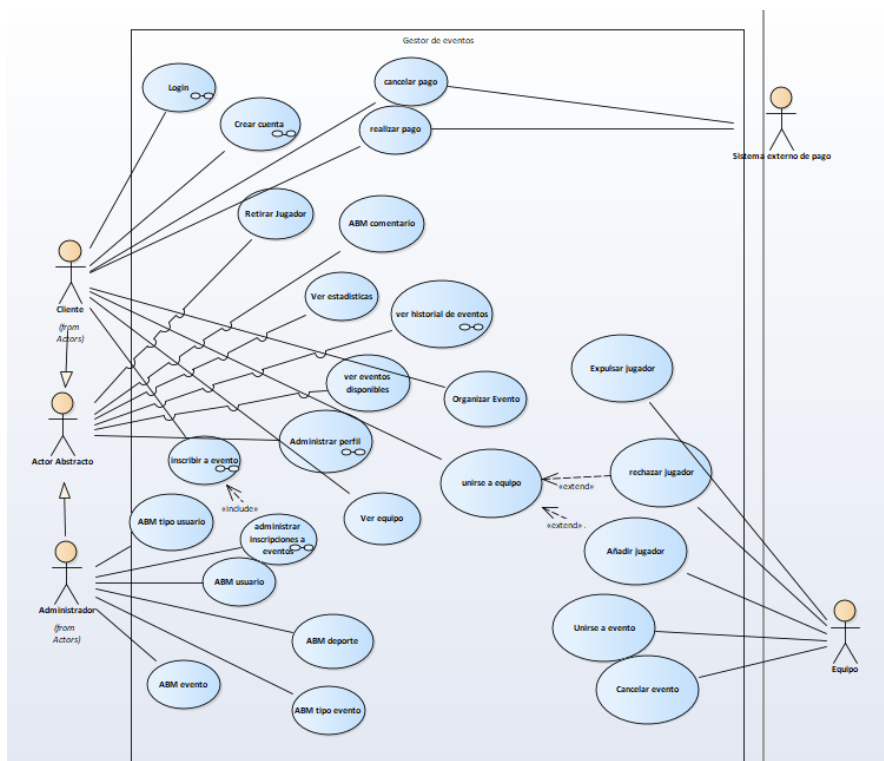


Figura 45: Módulo de casos de uso de playinga

2.3) Problemas y necesidades detectados en las funciones relevadas en detalle y en su entorno organizacional.

### Problemas

- No posee servicio de mensajería.
- La aplicación no permite ver reportes financieros a los jugadores.
- Los eventos pueden ser solamente de 2 tipos, partidos o torneos.
- El sistema no permite participar en eventos sin equipo.
- Limitación de los históricos de partidos y torneos.

### Necesidades

- Poseer un medio de comunicación rápido y eficaz dentro de la aplicación
- Como jugador necesito observar los históricos de transacciones que he hecho.
- Crear diferentes tipos de eventos parametrizables, como entrenamientos.
- Capacidad de participar en un evento, ya sea partido o torneo, como un jugador individual para deportes que lo requieran, sin necesidad de crear un equipo.
- Ampliar los filtros y alcance de los reportes de partidos y torneos en los que participó un jugador en particular.

## SportsPlus



Figura 46: Logo SportsPlus

### 1) Relevamiento general

#### 1.1) De la organización

SportsPlus es una plataforma de gestión deportiva moderna todo en uno para clubes deportivos, ligas, asociaciones, equipos, entrenadores, administradores, oficiales, jugadores y padres. La plataforma SportsPlus ayuda con la gestión de registros de jugadores, pagos en línea, horarios, equipos, ligas, torneos, campos, miembros y sitio web.

#### 1.2) Funciones detectadas

Algunas de las funciones detectadas de la aplicación SportsPlus son:

- **Lista de equipos:** Crea y gestiona listas de equipos con todos los detalles del jugador en un solo lugar.
- **Calendario del equipo:** Juegos, prácticas y eventos en equipo con recordatorios y notificaciones instantáneas. Detecta conflictos y evita confusiones.
- **Disponibilidad del jugador (RSVP):** Permite a los jugadores actualizar su disponibilidad. El entrenador y el personal del equipo pueden ver quién viene y quién no.
- **Live:** Comparte puntajes, imágenes, mensajes y cualquier actualización de última hora en tiempo real junto con la notificación móvil.
- **Gestión de voluntarios:** El personal del equipo puede abrir puestos de voluntariado. Los miembros del equipo pueden inscribirse para ser voluntarios.
- **Recursos:** Carga y comparte cualquier recurso, incluidos documentos, imágenes y videos con el equipo.
- **Anuncios:** Publica y comparte anuncios con el equipo.
- **Imágenes:** Carga y comparte imágenes con todo el equipo.
- **Mensajería del equipo:** Envío de mensajes a todo el equipo (correo electrónico, notificación móvil, mensaje en línea, mensaje en vivo).
- **Resultados:** Actualización de los resultados del juego con la clasificación de la liga y el resumen del juego.
- **Panel de control de administración:** Los miembros del personal del equipo pueden configurar las opciones de accesibilidad, seguridad y privacidad para los miembros del equipo.
- **Miembros de la familia:** Fácil cambio entre cuentas de miembros de la familia y acceso a todos los equipos, horarios, mensajes de los miembros de la familia ...
- **Creador de sitios web:** sitio web integrado para la organización.
- **Registros:** Abrir el registro en línea.
- **Cobrar pagos en línea.**
- **Programación:** Juegos, prácticas, eventos.
- **Liga y gestión de torneos.**
- **Gestión de instalaciones:** Incluidas reservas.
- **Recaudación de fondos:** Tienda, donaciones, patrocinios.
- **Gestión de membresía:** Atletas, personal, padres, funcionarios.
- **Comunicación:** correo electrónico, en línea, SMS, notificaciones móviles.
- **Portal del jugador y gestión de miembros de la familia.**

### 1.3) Tecnología de información

Las tecnologías utilizadas en Eventols son:

#### **Sistema operativo**

- Todos (multiplataforma)

#### **Base de datos**

- No especifica

#### **Lenguaje de programación**

- CSS
- HTML
- JavaScript
- Bootstrap

#### Otras herramientas

- React
- Nodejs
- Redux
- DataTables
- Modernizr
- Select2
- SweetAlert2
- Google Analytics
- Google Tag Manager
- Moment.js

#### Versión móvil

- [Android](#)
- [Apple](#)

2) Relevamiento detallado y análisis del sistema.

2.1) Detalle, explicación y documentación detallada de todas las funciones seleccionadas.

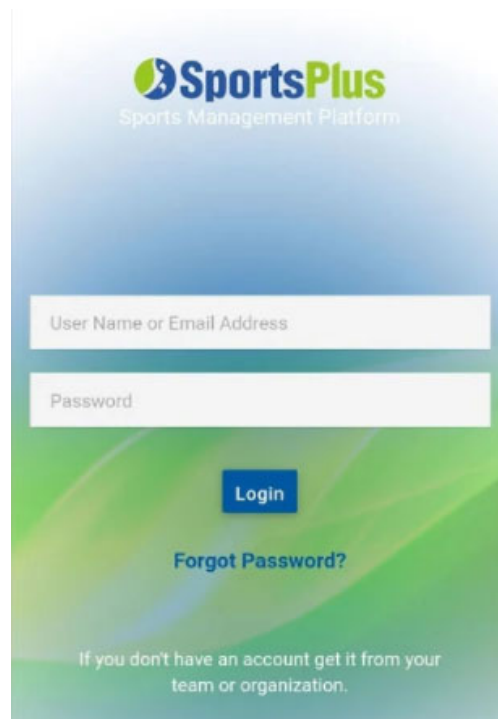


Figura 47: Interfaz de inicio de sesión de SportsPlus

**Módulo de seguridad:** El módulo de seguridad se encarga de pedir el email y la contraseña al usuario que desea ingresar al sistema, una vez iniciada sesión dependiendo del rol asignado se le habilitan ciertos tipos de menús.

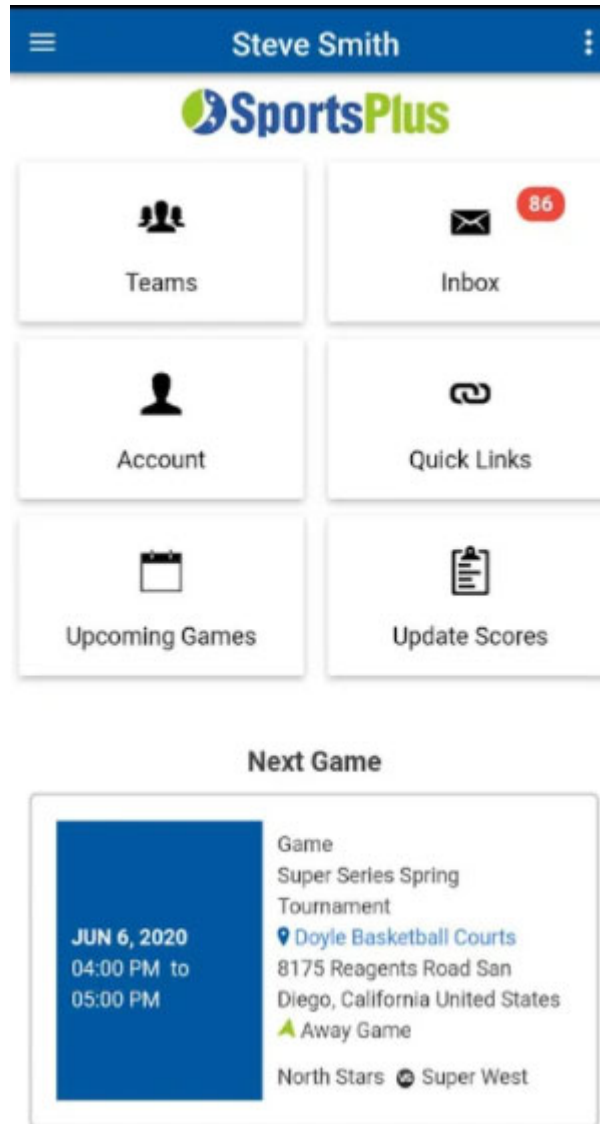


Figura 48: Módulo de seguridad de SportsPlus

**Módulo de perfil:** Es el encargado de administrar el perfil del usuario logueado. Le proporciona de una forma rápida y simple información importante, como por ejemplo el próximo juego, los mensajes recibidos, los resultados de otros partidos.




Steve Smith	
<b>Update Scores</b>	
<b>Mar 14 2020</b> 09:15 AM to 10:15 AM	Game  Super Series Spring Tournaments Pauley Pavilion by Wescom 301 Westwood Plaza Los Angeles, California United States Home Game Young Bullz vs Good Hearts Result: <b>Win</b> Score: 38 - 35
<b>Mar 14 2020</b> 08:15 AM to 09:15 AM	Game  Super Series Spring Tournaments Mission Creek Center Arena Los Angeles, California United States Home Game Good Hearts vs Young Guns Result: <b>Loss</b> Score: 35 - 41
<b>Mar 10 2020</b> 08:15 AM to 10:15 AM	Game  Super Series Spring Tournament Skyport Plaza Building 580, University of California Santa Barbara, Stadium Rd, California United States Home Game

Figura 49: Perfil de SportsPlus

**Módulo Calendario:** Tiene la responsabilidad de administrar y gestionar todos los eventos de los equipos o usuarios, cuando el usuario los agrega, los guarda y genera reportes y alertas, como recordatorios para asistir al usuario.

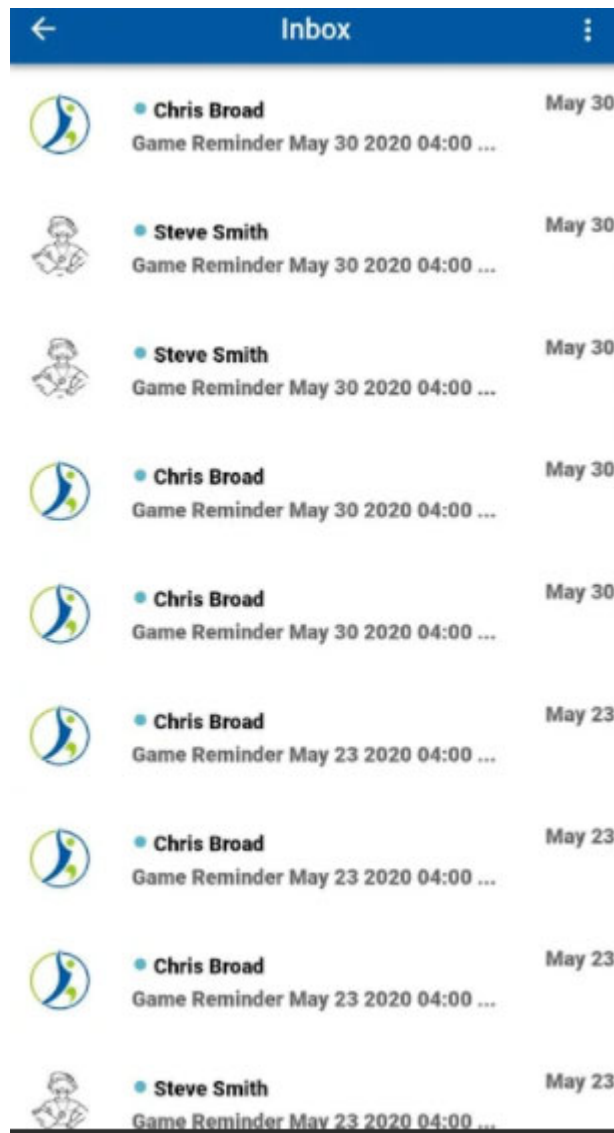


Figura 50: Calendario en SportsPlus

**Módulo de notificaciones:** Este módulo es el encargado de mandar notificaciones al celular de los participantes de los eventos, también puede configurar alerta para equipos por los que simpatiza.

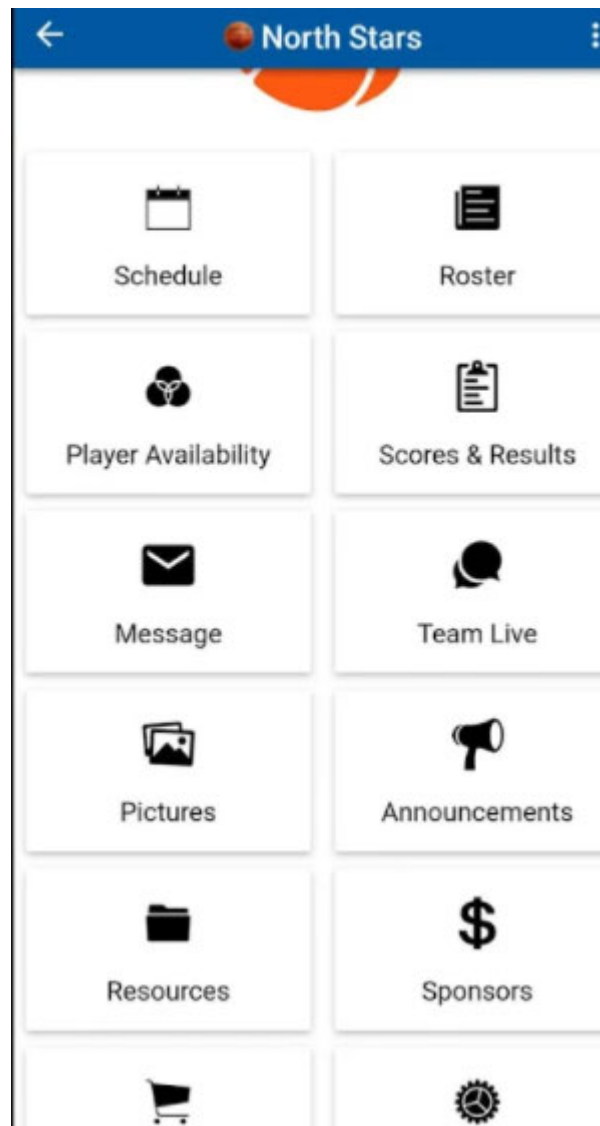


Figura 51: Módulo de notificaciones de SportsPlus.

**Módulo del perfil del equipo:** En este módulo se le muestra al administrador un pequeño dashboard, con menús donde se puede ver información del equipo, además puede hacer clic en los distintos ítems, para por ejemplo, ir al módulo de publicidad o al módulo de sponsors.



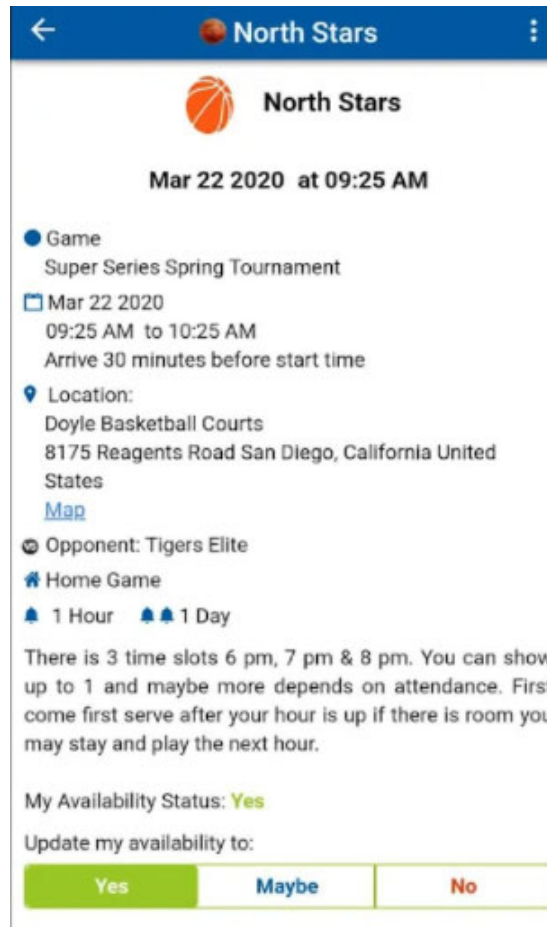


Figura 52: Perfil del equipo de SportsPlus.

**Módulo de información del evento:** Muestra información del evento configurado con anterioridad, además tiene una funcionalidad interesante, si participamos del evento podemos elegir la “disponibilidad” para el evento, lo cual permite al coordinador del evento tener la información completa sobre quien asiste o no al evento.



Figura 53: Módulo de información del evento de SportsPlus

**Módulo mensajería:** Es el módulo encargado de brindar un servicio de mensajería entre distintos usuarios, ya sean miembros del propio equipo o entrenadores. Puede ser entre dos usuarios o entre varios usuarios simultáneamente, utilizando la función de chat grupal.

## 2.2) Modelo lógico del sistema actual.

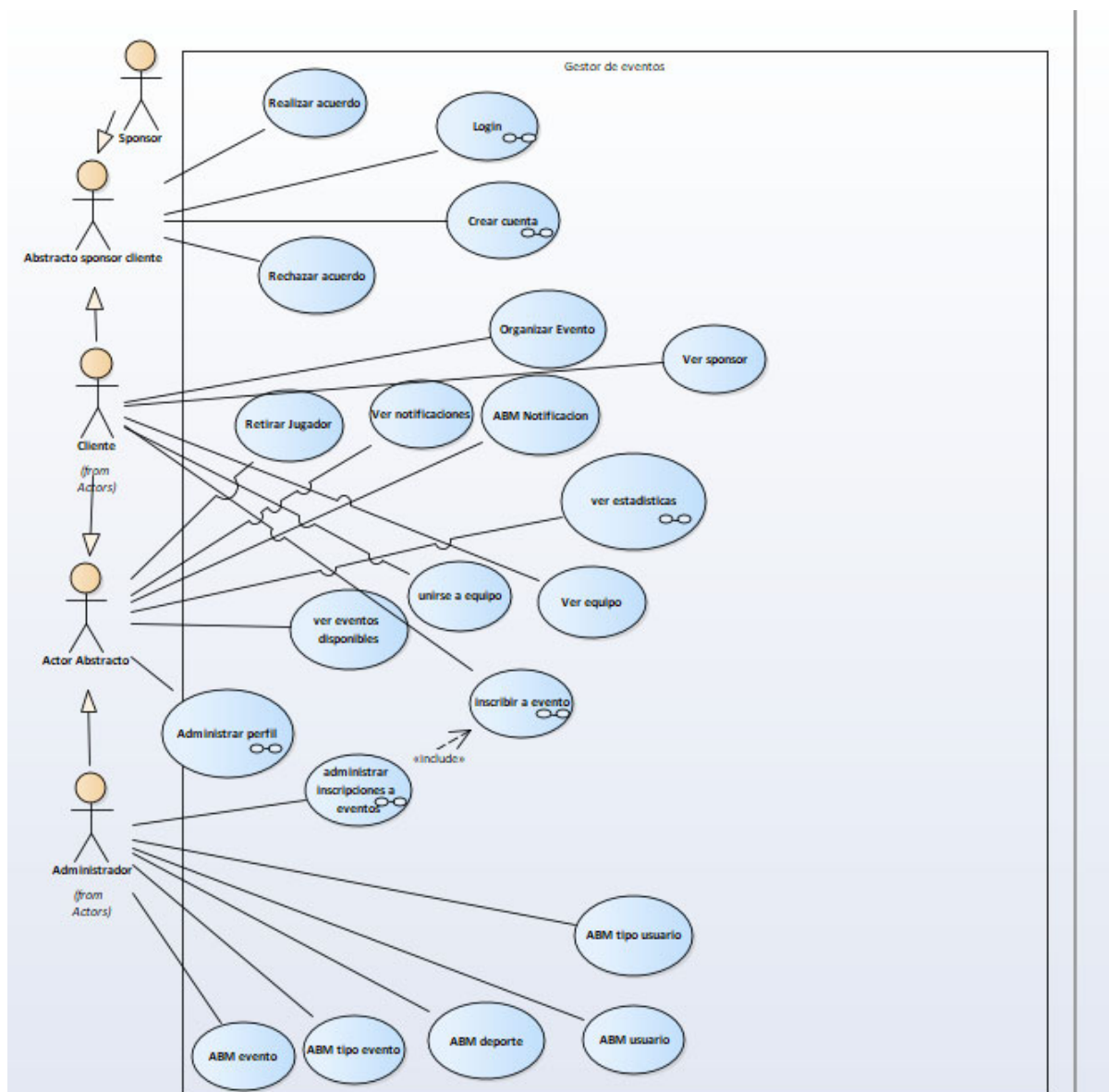


Figura 54: Diagrama de casos de uso de SportsPlus

## 2.3) Problemas y necesidades detectados en las funciones relevadas en detalle y en su entorno organizacional.

### Problemas

- No es posible gestionar los pagos de los eventos dentro de la aplicación.
- No soporta calificaciones y relevancia de equipos, usuarios o sponsors.
- Las notificaciones se vuelven incontrolables en cierto punto, al relacionarse con diferentes eventos y usuarios.

### Necesidades

- Se requiere la posibilidad de gestionar las transacciones de pagos para asistir a los eventos.
- Establecer un sistema de gamificación o de puntaje y relevancia para los diferentes tipos de usuarios, añadiendo funcionalidades como comentarios y puntuaciones.
- Gestionar las notificaciones de manera tal que el usuario pueda seleccionar las más relevantes y separarlas del resto, como así también, la capacidad de elegir la forma en que se presentan.

### 2.3 Cuadro resumen de las funciones detectadas en los sistemas relevados

					
Estadísticas en tiempo real	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Inscripción Online	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Pagos Online	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Formulario de inscripción customizado	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Deportes múltiples	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Distintos tipos de torneos	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Reportes generales del evento	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Personalización del equipo armado	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Mensajería en la aplicación	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Alineación del equipo	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PostPartido	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Gamificación	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Navegación	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Publicidad	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Versión Móvil	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Gratis	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

## 2.4 Objetivos y alcances preliminares del nuevo Sistema.

### 2.4.1 Objetivos preliminares del sistema:

El objetivo del sistema es proveer una herramienta que facilite la gestión de eventos deportivos y que motive a la generación de más eventos de este tipo. Para esto, el sistema busca satisfacer las necesidades de todas las entidades que pueden participar en un evento deportivo, como pueden ser, el organizador, los deportistas, los patrocinadores y hasta incluso las necesidades de los árbitros del evento. Además, busca motivar a los deportistas a participar de los eventos existiendo la posibilidad de estar entre los deportistas destacados de la aplicación y ser visto por patrocinadores.

## 2.4.2 Tecnologías a utilizar en el desarrollo

### Front End

- Lenguaje: TypeScript
- Framework: Angular
- HTML, CSS

### Base de Datos:

- MongoDB
- Redis (para los datos no persistentes)

### BackEnd:

- API REST
- Lenguaje: JavaScript
- Framework: Nodejs 14 (modular) ECMAScript 6
- Docker
- Postman

### Gestión:

- Gestor de Configuración: GitHub
- Kanban: GitHub Project Board
- Google Drive

## 2.4.3 Alcances preliminares del nuevo sistema

### Módulo de seguridad

El módulo de seguridad se encarga de asignar permisos a los distintos roles de la aplicación. También comprende el servicio de autenticación y creación de cuentas, utilizando el método de usuario y contraseña. Los tipos de cuentas son:

- Organizador: La cuenta de organizador posee permisos para:
  - Manejar una parte del módulo de eventos, como la de creación/modificación/baja de un evento.
  - Manejar el módulo de publicidad, aceptando o rechazando los distintos patrocinadores que desean participar del evento.
  - Usar el módulo de reporte.
  - Emplear el módulo de pagos, ofreciendo los métodos de pagos disponibles para el evento (también puede disponer formas de pago por defecto para todos los eventos).
- Patrocinador: La cuenta de patrocinador posee permisos para:
  - Ver en la pantalla principal los organizadores más sobresalientes, evaluados basándonos en la cantidad de eventos organizados, la cantidad de deportistas suscritos y su calificación promedio),

- Posibilidad de suscripción a una cuenta de organizador para estar al tanto de los eventos que este realiza.
  - Describir la forma en la cual desea patrocinar el evento
  - Indicar las condiciones que se deben cumplir para su participación.
- **Árbitro:** El árbitro posee permisos para:
  - Registrar el resultado de los encuentros
  - Registrar el pago de los equipos.
- **Club:** El Club posee permisos para:
  - Aceptar/rechazar eventos.
  - Informar ubicación del lugar físico donde dará soporte a los eventos aceptados.
  - Ofrecer días y horarios disponibles.
  - Informar las canchas disponibles.
- **Deportista:** El deportista representará al usuario base del sistema, tendrá permisos disponibles para todos los demás roles, excepto el de árbitro, entre sus permisos encontramos:
  - Utilizar el módulo de inscripción para inscribirse a un evento en particular.
  - Suscribirse a un organizador, para estar al tanto de los eventos y publicaciones que este produzca.
  - Emplear el módulo de reporte respecto a sus participaciones en los eventos, por organizador.
  - Utilizar el módulo de pago y elegir entre los métodos disponibles
  - Ver en la pantalla principal los organizadores más sobresalientes en su provincia, evaluados basándonos en la cantidad de eventos organizados, la cantidad de deportistas suscritos y su calificación promedio.

Además, los usuarios podrán cambiar su email desde el sistema y confirmarlo desde el nuevo email seleccionado y podrán cambiar su contraseña desde:

- El apartado de Inicio de sesión, para solicitar un link, que le llegará a su email y desde ahí poder cambiar su contraseña.
- El perfil dentro del sistema, debiendo introducir su contraseña actual, para luego hacer el cambio.

Los usuarios podrán contactarse con el administrador a través del apartado “Contactanos”, toda vez que hayan creado su cuenta. Una vez enviado el mensaje, deberán esperar 4 horas para poder enviar otro.

### Módulo de eventos

El módulo de eventos es el encargado de gestionar todos los eventos deportivos dentro del sistema, desde dar de alta/baja/modificación a un evento. Hasta la asignación de las personas que participan del mismo, además es responsable de la integridad de los datos sobre los eventos dando una respuesta confiable y de manera eficiente.

Como *organizador*, el módulo ofrecerá la posibilidad de gestionar por completo un evento, permitiendo la modificación de cualquier información que pertenezca al evento, hasta el momento del inicio de las inscripciones al mismo. En el momento de la creación un evento, el organizador deberá elegir un club existente en el sistema o seleccionar una ubicación.

Se podrán visualizar los eventos creados según distintos criterios y filtros para facilitar al usuario la navegación dentro de los mismos.

Como *patrocinador*, permite ver toda la información de los eventos próximos, pudiendo evaluar el usuario, su patrocinio.

Como *club*, permite registrar las instalaciones en la plataforma, y así, presentarlas como ubicaciones para realizar eventos, permitiendo a los organizadores su exploración, y posterior selección, según sus necesidades.

### Módulo de reportes

El módulo reportes se encarga de administrar la presentación de la información referente a todas las entidades participantes de la aplicación que necesiten llevar algún tipo de seguimiento, mediante diversas formas de representación gráfica según lo requiera el usuario.

También debe definir los permisos de los distintos tipos de usuario para su acceso a estos reportes, protegiendo de esta manera la información sensible.

Además, este módulo es el encargado de efectuar los reportes de manera efectiva, con todos los datos necesarios para que la información sea íntegra y comprensible.

### Módulo de publicidad

El módulo publicidad se encarga de administrar y organizar todo lo referido a publicidad, facilitando la obtención de los ingresos, que permiten mantener el sistema funcionando, y a la vez, ofrecer al usuario la posibilidad de ver las publicidades.

### Módulo de inscripción

El módulo de inscripción es el encargado de dar altas/bajas/modificaciones, a las inscripciones de los participantes a los eventos, guardando esta información, de la manera más íntegra y óptima posible.

Es encargado de traer todas las inscripciones asociadas a un evento, ordenadas según los parámetros que corresponda.

Además, debe proveer toda la información referente a una inscripción en particular, como también invalidar una inscripción cuando sus datos no sean los correctos.

### Módulo de pagos

El módulo de pagos se encarga de gestionar las transferencias financieras, dentro de la plataforma, entre los diferentes usuarios, ya sean clientes, administradores, proveedores, clientes publicitarios, clubes, etc., otorgando la posibilidad de elegir la forma de pago, para luego coordinar con los servicios externos como Mercado Pago. Debe validar y verificar cada pago, su estado, y alertar sobre cualquier discrepancia con el acuerdo entre las partes, para cumplir con las inferencias legales que conciernen a la aplicación.

### Módulo de Feedback

El módulo de feedback es el encargado de gestionar el “post partido”, es decir, una vez realizado el evento cada jugador puede asignar una puntuación, realizar un comentario y subir fotos del evento. Para que un jugador pueda ejecutar el feedback tiene que haber participado en el evento, teniendo un tiempo límite de una semana, esto es para evitar fraudes.

### Módulo de parametrización

Debe lograr la parametrización del sistema de información en cada una de sus diferentes funcionalidades, adaptando la información, los objetos y los elementos a los parámetros establecidos previamente. Como por ejemplo, al seleccionar un deporte específico como el boxeo, el sistema debe dar por hecho que el equipo es de una sola persona permitiendo la restricción de parámetros de entrada en ese campo. Es el encargado de organizar y estandarizar la información ingresada en parámetros de entrada para ejecutar diferentes consultas y que estas sean fiables.

### Módulo Perfil

Se encarga de gestionar las operaciones que cada rol puede ejecutar en su perfil, y los datos que se van a visualizar en el mismo, mostrando información relevante del usuario.

Las operaciones básicas que puede realizar cada rol sobre su perfil, son entre otras:

- Modificar:
  - Nombre.
  - Imagen de perfil.
  - Descripción de la cuenta.
  - Darse de baja.

Habilita al usuario a modificar y mostrar los datos de su perfil hacia todos los usuarios del sistema que estén interesados con completa libertad.







## Diseño

### 1. Objetivos y alcances definitivos del nuevo Sistema.

#### Objetivo del Sistema

El objetivo del sistema es proveer una herramienta que facilite la gestión de eventos deportivos y motive a la generación de más eventos deportivos. Para esto, el sistema busca satisfacer las necesidades de todas las entidades que pueden participar en un evento deportivo, tales como el organizador, los deportistas, los patrocinadores, sin dejar de atender las necesidades de los árbitros del evento. Además, busca alentar a los deportistas a participar de los eventos ofreciendo la posibilidad de estar entre los deportistas destacados de la aplicación y así ser vistos por patrocinadores.

#### Alcance Funcional

##### Módulo de seguridad

El módulo de seguridad incluye el servicio de autenticación y creación de cuentas, utilizando el método de usuario y contraseña. Los tipos de cuentas son:

- Organizador: La cuenta de organizador posee permisos para:
  - Manejar una parte del módulo de eventos, como la parte de creación/modificación/baja de un evento.
  - Manejar el módulo de publicidad, aceptando o rechazando los distintos patrocinadores desean participar en el evento.
  - Usar el módulo de reporte.
  - Emplear el módulo de pagos ofreciendo los métodos de pago disponibles para el evento, y permitiendo disponer formas de pago por defecto para todos los eventos.
  - Registrar el resultado de los encuentros.
  - Registrar el pago de los equipos.
- Patrocinador: La cuenta de un patrocinador posee permisos para:
  - Observar en la pantalla principal los organizadores más destacados, evaluados basándonos en la cantidad de eventos organizados, la cantidad de deportistas suscritos y su calificación promedio.

- Suscribirse a la cuenta de un organizador para estar al tanto de los eventos que este realiza.
  - Indicar la forma en la cual desea patrocinar el evento.
  - Describir las condiciones que se deben cumplir para su participación.
- **Árbitro:** El árbitro posee permisos para:
  - Registrar el resultado de los encuentros
  - Registrar el pago de los equipos.
- **Club:** El Club posee permisos para:
  - Aceptar/rechazar eventos.
  - Indicar ubicación del lugar físico donde dará soporte a los eventos aceptados.
  - Informar los días y horarios disponibles.
  - Indicar las canchas disponibles.
- **Deportista:** El deportista representa al usuario base del sistema, sus permisos están disponibles para todos los demás roles, excepto el de árbitro. Entre sus permisos encontramos:
  - Utilizar el módulo de inscripción, y así inscribirse a un evento en particular
  - Suscribirse a un organizador para estar al tanto de los eventos y publicaciones que este haga.
  - Utilizar el módulo de reporte, respecto a sus participaciones en los eventos, filtrando por organizador.
  - Usar el módulo de pago eligiendo entre los métodos disponibles.
  - Ver en la pantalla principal los organizadores destacados en su provincia (evaluados con base en la cantidad de eventos organizados, la cantidad de deportistas suscritos y su calificación promedio).
- **Cuenta Administrador del Sistema:**
  - Puede realizar configuraciones sobre:
    - Tipos de organizadores.
    - Estado de inscripción de un evento.
    - Estado del evento.
    - Estado del patrocinador del evento.
    - Tipos de deportes.
    - Tipos de eventos.

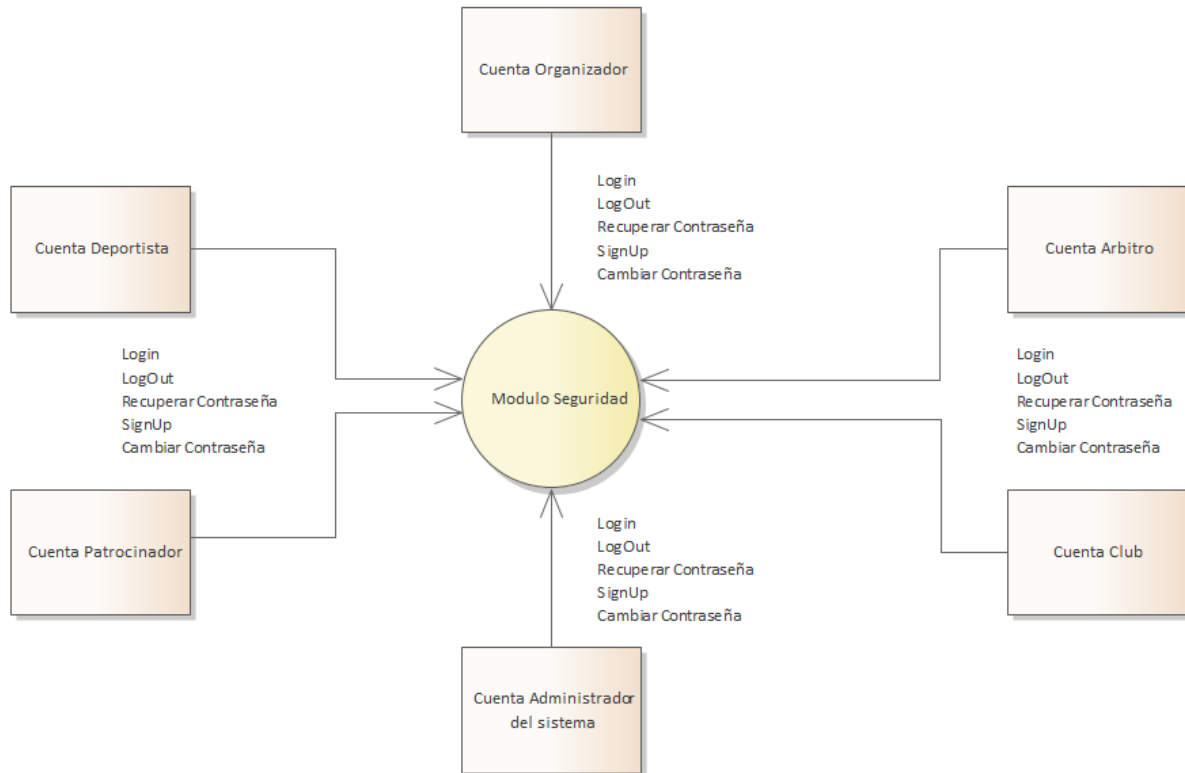


Figura 55: Diagrama de contexto del módulo de seguridad

### Módulo de eventos

El módulo de eventos es el encargado de gestionar todos los eventos deportivos dentro del sistema, como dar de alta/baja/modificación a un evento, o la asignación de las personas al mismo. Se encarga de la integridad de los datos sobre los eventos y da información confiable, en tiempo real y de manera eficiente.

#### Alta Evento

A cada *organizador* el módulo le ofrecerá la posibilidad de crear un evento, indicando las siguientes características, y en este orden:

1. Indicar el nombre del evento.
2. Indicar Tipo de Evento.
  - a. Liga:
    - i. Indicar Clubes en donde se realizarán los partidos.
  - b. Torneo
    - i. Indicar Club en donde se efectuarán los partidos.
3. Indicar categorías del evento.
  - a. Nombre de la categoría (por defecto el mismo del evento)
  - b. Indicar días y horarios de juego disponibles de la categoría.
  - c. Indicar precio de inscripción a la categoría del evento.
  - d. Indicar cantidad máxima de equipos del evento.

- e. Indicar la cantidad mínima y máxima de jugadores en los equipos.

Una vez creado el evento se le notificará al/los clubes respecto a la petición del organizador para realizar el evento en sus instalaciones.

#### *Modificación Evento*

Como *organizador* el módulo ofrecerá la posibilidad de gestionar por completo un evento dependiendo del momento en que lo haga:

- Si es antes de la fecha de inicio de inscripción, puede:
  - Modificar el nombre del evento.
  - Agregar categorías.
  - Eliminar categorías.
  - Modificar el nombre de la categoría.
  - Modificar días y horarios disponibles de juego.
  - Modificar el precio de inscripción.
  - Modificar la fecha de inicio de inscripción.
  - Modificar la fecha final de inscripción.
  - Modificar la fecha de inicio del evento.
  - Modificar la fecha del final del evento.
  - Modificar la cantidad máxima de equipos.
  - Modificar la cantidad mínima y máxima de jugadores en los equipos.

Pasada la fecha de inscripción, este no podrá efectuar modificaciones, debido a que puede haber inscripciones comprometidas.

#### *Eliminar Evento*

El módulo ofrecerá la posibilidad al *organizador* de eliminar un evento, dependiendo del momento en que lo haga:

- Si es antes de la fecha final de inscripción y el estado del evento es “Propuesto”:
  - El evento se da de baja.
- Si es antes de la fecha final de inscripción y el estado del evento es “Inscripciones en curso”:
  - El evento se da de baja.
  - Se notifica a los usuarios involucrados (Club, Deportistas, Patrocinadores)

Pasada la fecha final de inscripciones, este no podrá eliminar el evento, debido a que puede haber pagos comprometidos.

#### *Interactuar con el Evento*

Todos los *usuarios* podrán ver los reportes del evento, que incluyen los siguientes datos:

- Cantidad de deportistas inscriptos.
- Promedio de calificación del evento.
- Mejores y peores feedback.
- Resultados de los partidos registrados hasta el momento.
- Lugar/es donde se disputará el evento.

Los *participantes* podrán ver la próxima fecha a jugar en el momento en que entra a la información del evento.

El *patrocinador* puede inscribirse al evento solicitando:

- La forma de publicidad que ellos desean.
- La remuneración o materiales que están dispuestos a dar.

Cuando envíe la solicitud, se le notificará al *organizador*, éste último podrá aceptarla o rechazarla.

A todo *club* se le permite registrar sus instalaciones en la plataforma, para presentarlas como ubicaciones para realizar eventos, de esta manera los organizadores, explorando, podrán seleccionar la que más se ajuste a sus necesidades.

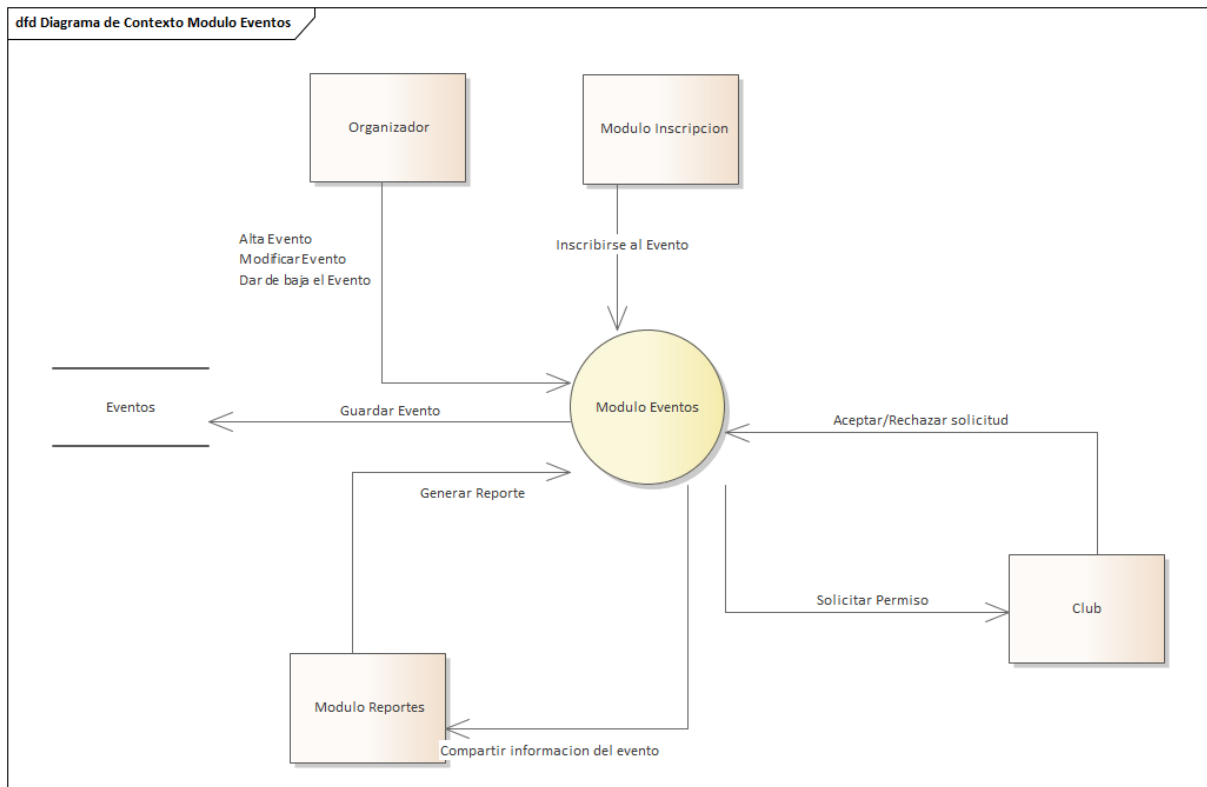


Figura 56: Diagrama de contexto del módulo de eventos

### Módulo de reportes

El módulo reportes se encarga de mostrar información relevante de cada ámbito:

- Eventos:
  - Calificación feedback promedio del evento.
  - Cantidad de Deportistas Inscriptos.
  - Mejores jugadores del evento.
- Organizador:
  - Calificación feedback promedio de todos los eventos del organizador.
  - Suscripciones filtradas por fechas.

- Cantidad de eventos filtrados por fecha.
- Cantidad de inscriptos por fechas.
- Patrocinador:
  - La cantidad de eventos patrocinados.
  - Las marcas ofrecidas en su perfil.
  - Eventos patrocinados anteriormente.

El módulo reportes se encarga también de administrar la presentación de la información referente a todas las entidades de la aplicación que necesiten llevar algún tipo de seguimiento, mediante diversas formas de representación gráfica según lo requiera el usuario.

También debe definir los permisos de los distintos tipos de usuario para el acceso a estos reportes, protegiendo así la información sensible, de los usuarios que no deben tener acceso a esta.

Además, este módulo es encargado de realizar los reportes de manera efectiva, con todos los datos necesarios para que la información sea íntegra y comprensible.

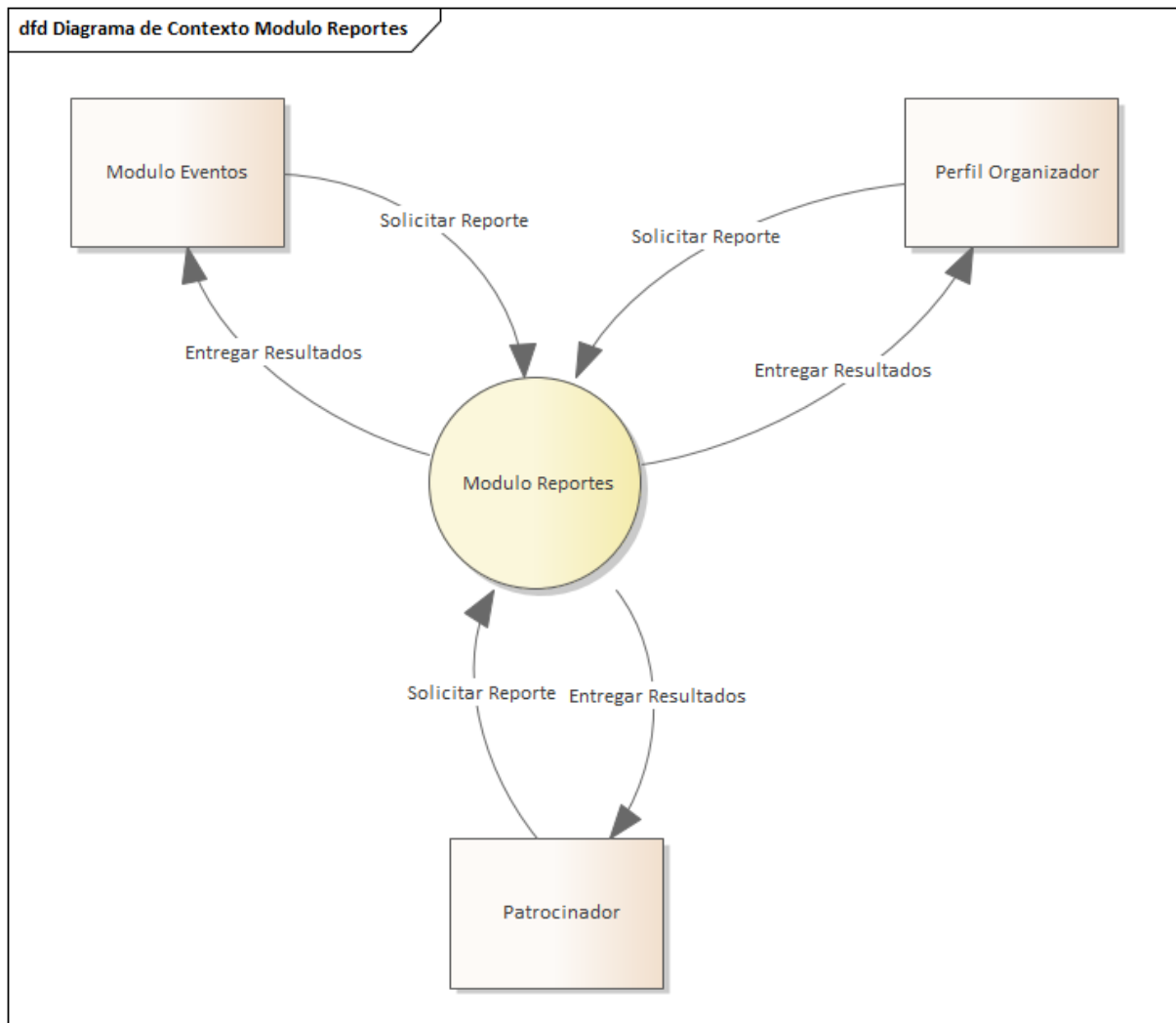


Figura 57: Diagrama de contexto del módulo de reportes

#### Módulo de publicidad

El módulo publicidad se encarga de administrar y organizar todo lo referido a publicidad como por ejemplo:

- Brinda la posibilidad de contactar por email, u otro tipo de contacto, con quienes administran las publicidades en la página, y de esta manera quienes estén interesados en este servicio, puedan concretar acuerdos publicitarios.
- Gestiona los anuncios publicitarios y sus características, como: los espacios en los que la publicidad tendrá lugar dentro de la plataforma y el tamaño de la misma, según lo pactado con el proveedor.

Cabe destacar que este módulo tiene una acción manual, es decir, el administrador del sistema será el encargado de subir las imágenes/gifs que se mostrarán en la página, luego de pactar con el interesado en la publicidad.



El administrador del sistema configura la publicidad para que se despliegue en un sector de la página, permitiendo colocar el aviso donde y como se haya pactado.

#### Módulo de inscripción

El módulo de inscripción es el encargado de dar las altas/bajas a las inscripciones de los participantes a los eventos, cuidando la integridad y optimizando el guardado de los datos. Debe traer todas las inscripciones asociadas a un evento, ordenadas según los parámetros que corresponda.

Además, debe garantizar la provisión de toda la información que posee una inscripción particular, invalidando una inscripción cuando sus datos sean incorrectos.

#### *Inscribirse a un evento*

Para que un usuario pueda inscribirse en un evento deberá:

- Seleccionar el evento:
  - Para poder dar de alta la inscripción se deben cumplir las siguientes condiciones en el evento seleccionado:
    - El evento debe tener el estado “Inscripciones en curso” asociado.
    - El número máximo de participantes en el evento no debe haber sido alcanzado.
    - La fecha límite de inscripción del evento tiene que ser mayor o igual a la fecha actual.
    - El deportista debe cumplir con las siguientes características:
      - El deportista debe tener un equipo en caso de que sea un deporte de equipos.
      - No debe estar inscripto en el evento previamente.
- Una vez seleccionado el evento y cumplidas las condiciones anteriores el usuario debe presionar el botón “Continuar inscripción”, desde la vista del evento seleccionado.
- El deportista ahora tiene la posibilidad de elegir un horario para jugar sus partidos:
  - El usuario puede elegir un horario para todo el evento presionando “Configurar preferencia de horario”.
- El deportista tiene la posibilidad de elegir un medio de pago para el evento, siempre y cuando el evento lo soporte.
  - Para ello el deportista elegirá de un combobox, el medio de pago de su preferencia, efectivo o Mercado Pago.
- Luego de haber completado todas las opciones el usuario deberá presionar en “Inscribirme”.
- A continuación el usuario podrá observar su inscripción efectuada en la página del evento, la inscripción se crea con el estado “No Pago”.

#### *Darse de baja a un evento*

Para dar de baja un evento el deportista deberá ir a la página del evento en cuestión:

- Una vez en la página del evento, el usuario deberá presionar el botón “dar de baja”
- Una vez seleccionado se le mostrará un área de texto para que pueda colocar el motivo de su baja.
- Cuando el cuadro de texto esté ya completado el usuario deberá presionar en “Eliminar inscripción”
- Luego de bajarse de la inscripción, el número de participantes del evento disminuirá en 1 y el estado de la inscripción cambiará a “Inactivo”.

Para poder realizar esta acción deben cumplirse ciertas condiciones:

- El evento tiene que tener asociado el estado “Inscripciones en curso”.
- El inicio del evento debe ser posterior a dos días de la fecha actual, esto para ofrecer la inscripción a otro participante de emergencia.
- El usuario debe estar inscripto al evento previamente.

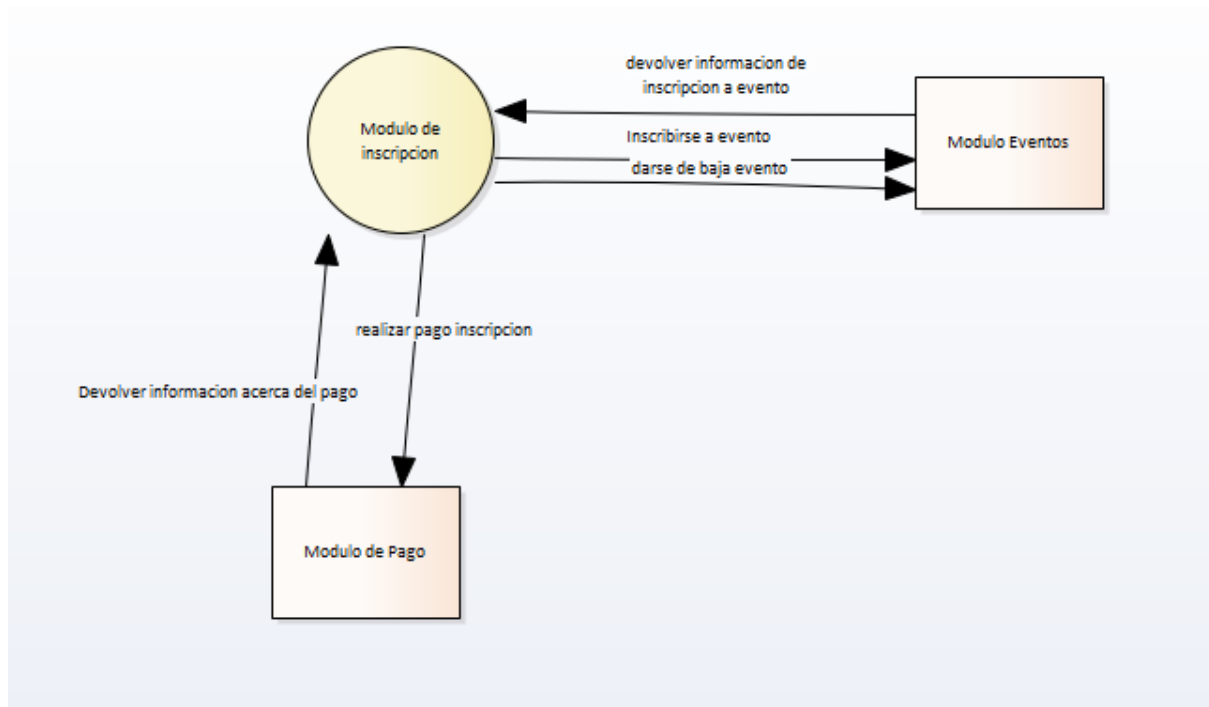


Figura 58: Diagrama de contexto del módulo de inscripción

### Módulo de pagos

El módulo de pagos se encarga de gestionar las transferencias financieras de la plataforma entre los diferentes usuarios de la aplicación, ya sean clientes, administradores, proveedores, clientes publicitarios, clubes, etc. También ofrece la posibilidad de elegir la forma de pago y coordinar con los servicios externos como Mercado Pago. Debe validar y verificar cada pago y el estado de estos y alertar sobre cualquier discrepancia con el

acuerdo entre partes, para cumplir con las inferencias legales que conciernen a la aplicación.

El módulo de pago guardará cada transacción realizada en nuestra base de datos, validando con el sistema externo primero, si la transacción realizada fue exitosa, y en qué estado se encuentra.

Cada deportista tiene la posibilidad de pagar el evento con su cuenta de Mercado Pago o en dinero en efectivo directamente al organizador, esto deberá ser registrado en la plataforma por su parte.

### Pago digital

En caso de que el pago sea digital, procederemos a utilizar la API de Mercado Pago la cual viene preparada para la gestión de pagos. Así mismo, una vez ejecutada la transacción digital se guardará en nuestra base de datos el estado de la inscripción.

La **API de Mercado Pago** funciona de la siguiente manera:

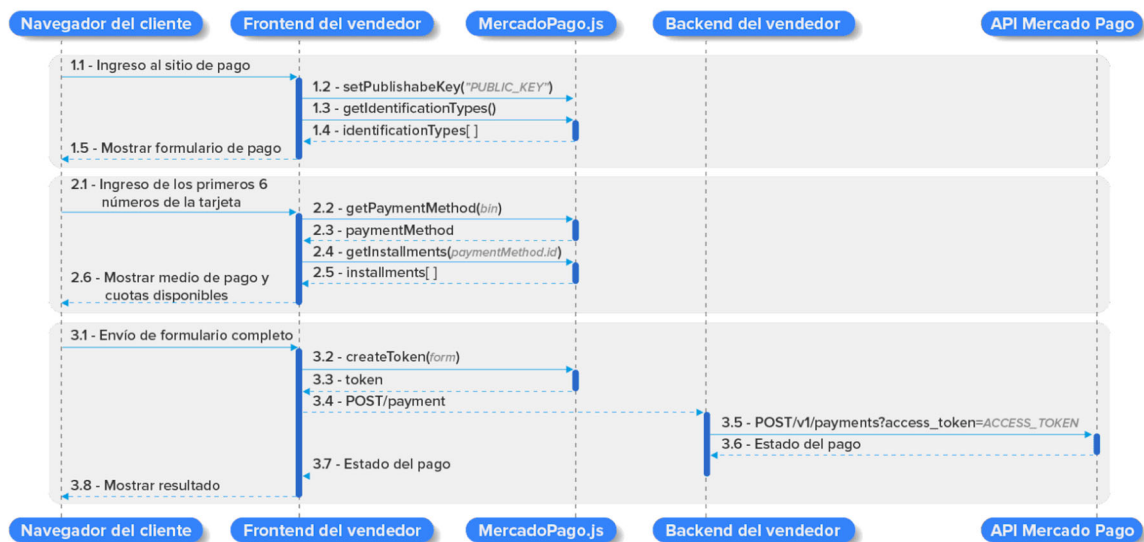


Figura 59: Diagrama de secuencia API Mercado Pago

De esta forma, nuestro frontend se encarga de preparar el pago del vendedor ofreciéndoles todas las funcionalidades y opciones de Mercado Pago. Una vez hecho esto, cuando se confirma el envío del formulario, la información llega a nuestro servidor y el backend se encarga de enviar el pago a Mercado Pago, esperar la respuesta y guardar el estado de la transacción final en la base de datos. A continuación, el backend da respuesta al front con el resultado de la transacción, para que sea visualizada por el usuario que comenzó el proceso de pago.

### *Pago en efectivo*

Si el pago es en efectivo, deberá ser registrado por el organizador del evento, o por el/los árbitros designados por el organizador el día del evento de manera presencial, haciéndose responsables de recolectar la suma exacta y de las personas correspondientes registradas.

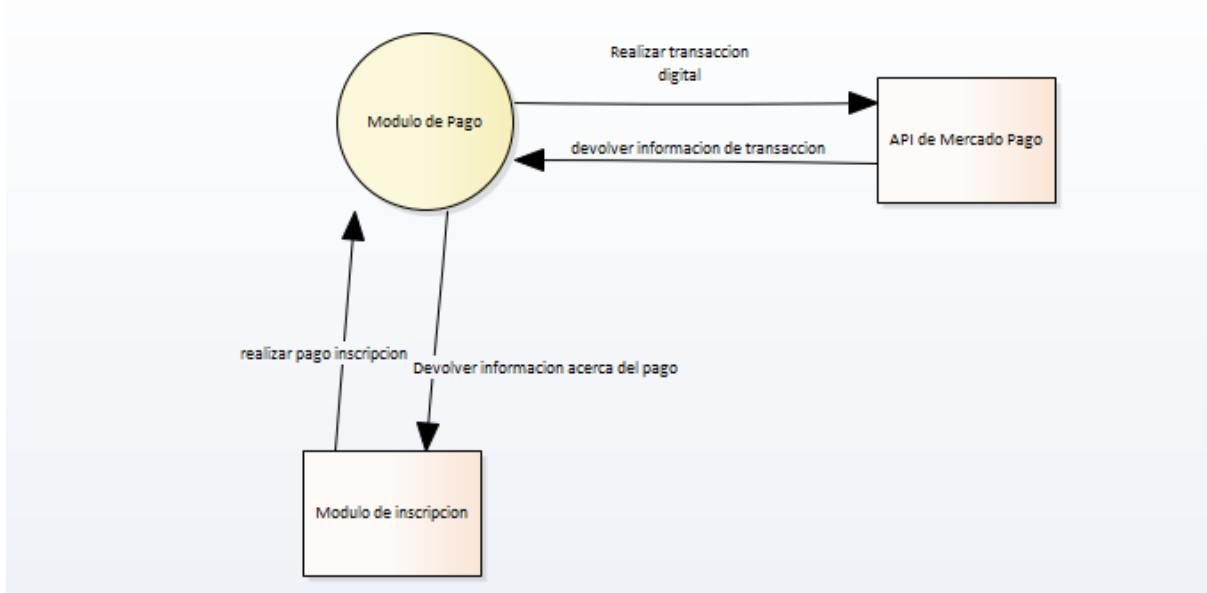


Figura 60: Diagrama de contexto del módulo de pagos

### Módulo de Feedback

El módulo de feedback es el encargado de gestionar el “post partido”, es decir una vez realizado el evento, cada jugador puede asignar una puntuación, realizar un comentario y subir fotos del evento. Para que un jugador pueda hacer el feedback tiene que haber participado en el evento, y tiene un tiempo límite de una semana para poder efectuar el feedback, esto es para evitar fraudes.

### *Dar de alta un feedback*

Para dar de alta un feedback el usuario deberá:

- Seleccionar un evento:
  - Condiciones que deben cumplirse en el evento seleccionado:
    - El evento tiene que tener asociado el estado “finalizado”.
    - El usuario debe cumplir con al menos una de las siguientes características:
      - Debe haber participado en el evento como deportista.
      - Debe haber participado en el evento como árbitro.
      - Debe haber participado en el evento como organizador.

- Una vez seleccionado el evento y cumplidas las condiciones anteriores el usuario debe presionar en “Dar feedback” desde la vista del evento seleccionado.
- El usuario ahora tiene varios tipos de feedback para elegir, puede realizarlos todos en conjunto o elegir una sola opción:
  - Dejar un comentario simple, presionando en “Agregar comentario”.
  - Subir una o varias fotos, presionando en “Adjuntar fotos”.
  - Valorar el evento, presionando en la opción “Valorar evento”, donde se desplegará una escala del 1 al 5, siendo 1 la peor valoración y 5 la mejor valoración.
- Luego de completar la/las opciones, el usuario deberá presionar en “Enviar feedback”.
- A continuación el usuario podrá observar su feedback en la página del evento, y será visible para todos los usuarios que entren a la vista.

#### *Dar de baja un feedback*

- Para dar de baja un feedback el usuario deberá ir a la página del evento seleccionado:
  - El evento tiene que tener asociado el estado “finalizado”.
  - El usuario debe haber participado con al menos uno de los siguientes roles:
    - Deportista.
    - Árbitro.
    - Organizador.
  - El usuario debe tener al menos un feedback publicado en la página del evento
- En la página del evento, el usuario deberá seleccionar su feedback en la lista de feedbacks del evento.
- Se desplegará su feedback.
- Para eliminar el feedback el usuario deberá presionar en “Eliminar feedback”.

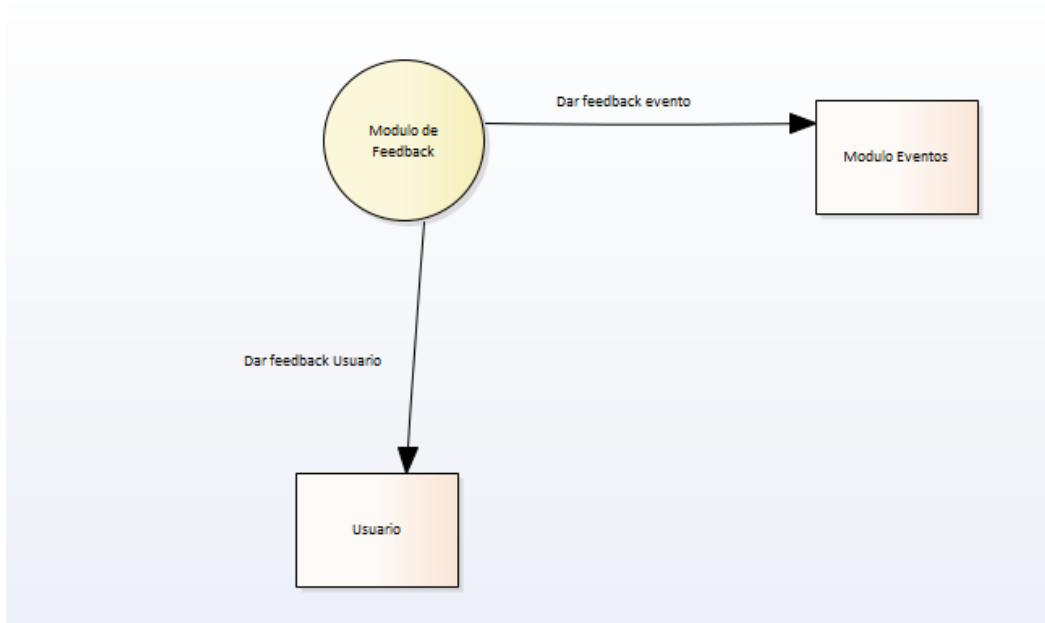


Figura 61: Diagrama de contexto del módulo de feedback

### Módulo de parametrización

Debe lograr la parametrización del sistema de información en cada una de sus diferentes funcionalidades, adaptando la información, los objetos y los elementos a los parámetros establecidos previamente. Como por ejemplo, al seleccionar un deporte específico como boxeo, el sistema debe dar por hecho que el equipo es de una sola persona permitiendo la restricción de parámetros de entrada en ese campo.

Es el encargado de organizar y estandarizar la información ingresada, en parámetros de entrada, para realizar diferentes consultas y que estas sean fiables.

#### *Dar de alta un parámetro*

Para dar de alta una parametrización, el usuario administrador del sistema debe elegir qué parámetro ingresar en la base de datos, teniendo la posibilidad de parametrizar las distintas tablas a través de un caso de uso específico "ABM" para el tipo de parámetro deseado.

- Tipo de Organizador: Premium, Standard, etc.
- Estado de Inscripción en el Evento: Inscrito, Desinscrito, etc.
- Estado del Evento: Inscripción Abierta, Inscripción Cerrada, En progreso, Terminado, Creado.
- Estado del Evento de Patrocinador: Patrocinado, No Patrocinado.
- Estado del Evento de Equipo: Equipo Inscrito, Equipo pendiente.
- Tipo de Deporte: Fútbol, Tenis, Básquet, etc.

- Localidad: Dorrego, etc.
- Provincia: Mendoza, etc.

En este módulo también se pueden dar de alta diferentes parámetros que no necesariamente estén relacionados con la lógica del negocio, como distintas configuraciones que hacen al funcionamiento de la aplicación:

- Dirección de la base de datos.
- Puerto de la base de datos.
- Nombre de usuario de la base de datos.
- Contraseña de la base de datos.
- Dirección de la API del backend.
- Lapso de tiempo entre cada backup.

### *BackUp*

Este módulo ofrece la opción de hacer un backup completo del sistema. El administrador podrá realizar copias de seguridad de la base de datos cuando el lo desee, a través de una interfaz gráfica, en el módulo de parametrización. También podrá restaurar dicho backup a través de una interfaz amigable para el usuario.

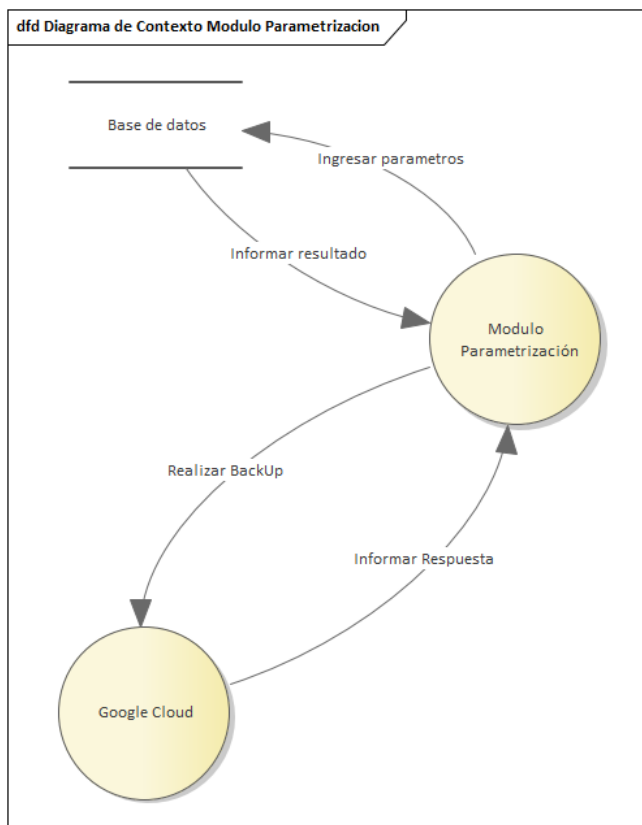


Figura 62: Diagrama de contexto del módulo parametrización

### Módulo Recomendación

Se encarga de ofrecer a los deportistas y patrocinadores, un conjunto de organizadores que están próximos a ofrecer un evento cerca de su zona, esta recomendación se basa en la cantidad de eventos realizados por el organizador, calificación promedio del organizador, cantidad de deportistas suscritos y el tipo de cuenta que tiene el organizador. Dicha recomendación estará en la pantalla principal del sistema, tanto para los deportistas, como para los patrocinadores, cabe aclarar que el término zona, es configurable por el usuario, pudiendo este filtrar por localidad/provincia.

#### *Recibir las recomendaciones*

Para recibir las recomendaciones el usuario realmente no debe ejecutar ningún paso previo, solamente con loguearse en la plataforma e ir a la página principal, estará en condiciones de recibir las recomendaciones por parte del módulo de recomendaciones.

El módulo también se encarga de gestionar la configuración para hacer las recomendaciones, esto significa que el módulo, utiliza ciertos datos de la base de datos relacionados con cada usuario, para filtrar las recomendaciones, haciendo visibles los eventos que puedan interesar a cada usuario en particular.

#### *Parámetros usados en las recomendaciones*

Los parámetros que utiliza la configuración de recomendaciones son tomados directamente desde los datos del usuario:

- Para realizar las recomendaciones se tendrá en cuenta:
  - **Ubicación del usuario (provincia):** Los usuarios estarán más interesados en los eventos que se realicen cerca de su zona.
  - **Historial de eventos de los organizadores:** Se tendrá en cuenta la cantidad de eventos que tiene cada organizador, ofreciendo como prioridad los organizadores que más eventos hayan organizado.
  - **Calificación promedio del organizador:** Se tendrá en cuenta la valoración promedio de cada organizador, ofreciendo(a modo de recompensa) los eventos de los organizadores con mejor valoración en primer lugar.
  - **Preferencias:** El usuario puede elegir alguna preferencia por un deporte determinado, lo cual hace que tenga peso en el momento de ofrecerle recomendaciones



- **Tipo de cuenta del organizador:** Habrá mayores probabilidades de recomendación para aquellos organizadores que tengan una cuenta prémium.
- Luego, se valora el peso de cada parámetro. A modo de ejemplo le asignaremos un peso a los distintos parámetros:
  - Ubicación: 5 puntos.
  - Historial de eventos de los organizadores: 2 puntos.
  - Calificación promedio del organizador: 4 puntos.
  - Cantidad de deportistas inscriptos al evento: 1 punto.
  - Preferencias: 3 puntos.
  - Tipo Cuenta: 2 puntos.
- Finalmente, a través de una búsqueda en la base de datos y cruzando información, se elige mostrar en orden descendente, los eventos de acuerdo a sus puntajes, según los criterios elegidos anteriormente.

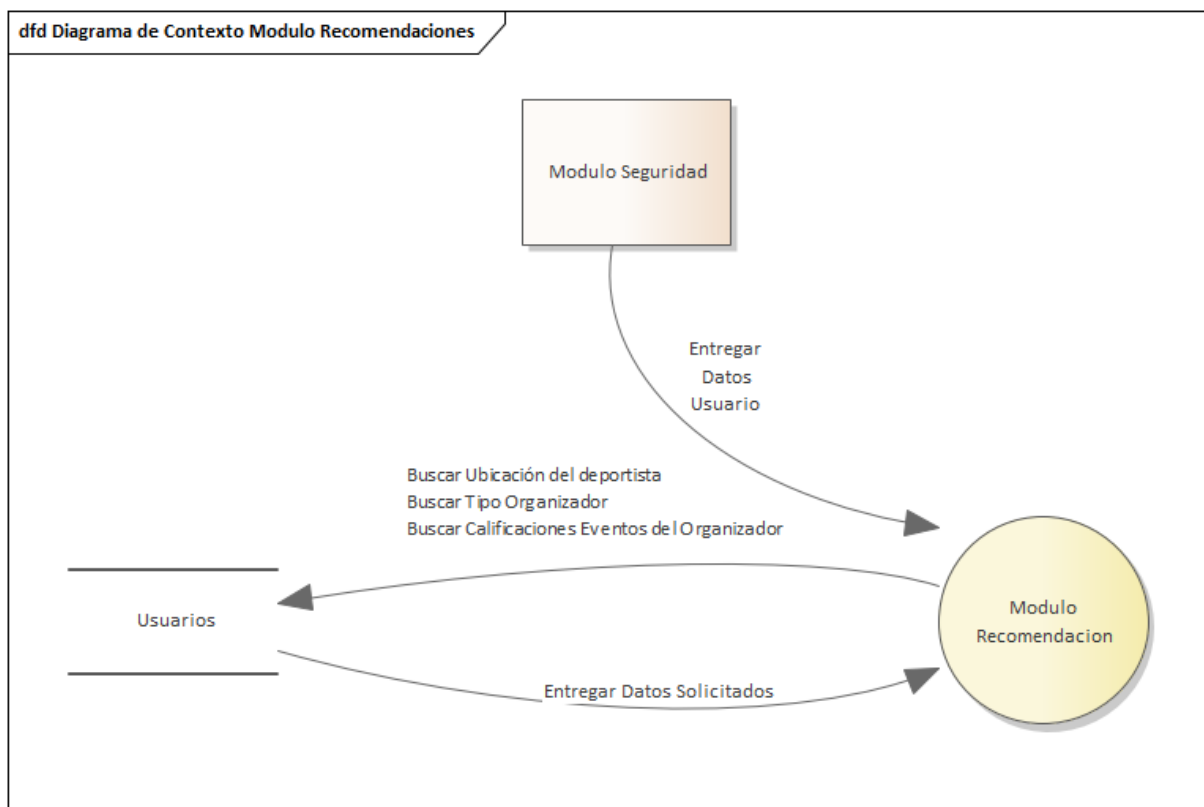


Figura 63: Diagrama de contexto del módulo de recomendaciones

## Módulo Sharing

Se encarga de gestionar la difusión de los elementos y entidades del sistema, brindando al cliente, una forma fácil de compartir los links en cualquier otra plataforma o red social, con un texto predefinido y el mismo link compartido, para que quienes vean esta publicación en las diferentes redes sociales en las que el usuario comparte, puedan ingresar al elemento o entidad del sistema con facilidad.

#### *Compartir un elemento*

Para compartir un elemento del sistema, el usuario deberá ir a la página del elemento que quiere compartir:

- Deberá seleccionar en la pantalla del elemento la opción “compartir”.
- Una vez seleccionada la opción, el usuario tendrá la posibilidad de elegir entre las distintas plataformas u opciones:
  - Compartir en:
    - Facebook.
    - Twitter
    - Instagram
  - Copiar vínculo del evento.

Los elementos que pueden compartir son:

- Evento
- Usuario
- Equipo

#### *Compartir un evento luego de crearlo*

Otra funcionalidad de este módulo es poder compartir un evento luego de que este fuese creado, sin la necesidad para el usuario de ir hasta la página del mismo y compartirlo manualmente:

- Para compartir un evento luego de crearlo, el usuario deberá contar con el rol de organizador, ya que este es el rol dentro de nuestro sistema que permite crear eventos.
- Luego, en la pantalla de creación del evento (explicada en el módulo de eventos), el usuario tendrá la posibilidad de compartir el evento, luego de que se haya generado, presionando en la opción “Compartir evento”.
- Si el usuario elige esta opción, una vez creado el evento se le desplegarán las diferentes opciones que tiene el sistema para compartir el evento:
  - Compartir en:
    - Facebook.
    - Twitter
    - Instagram
  - Copiar vínculo del evento.

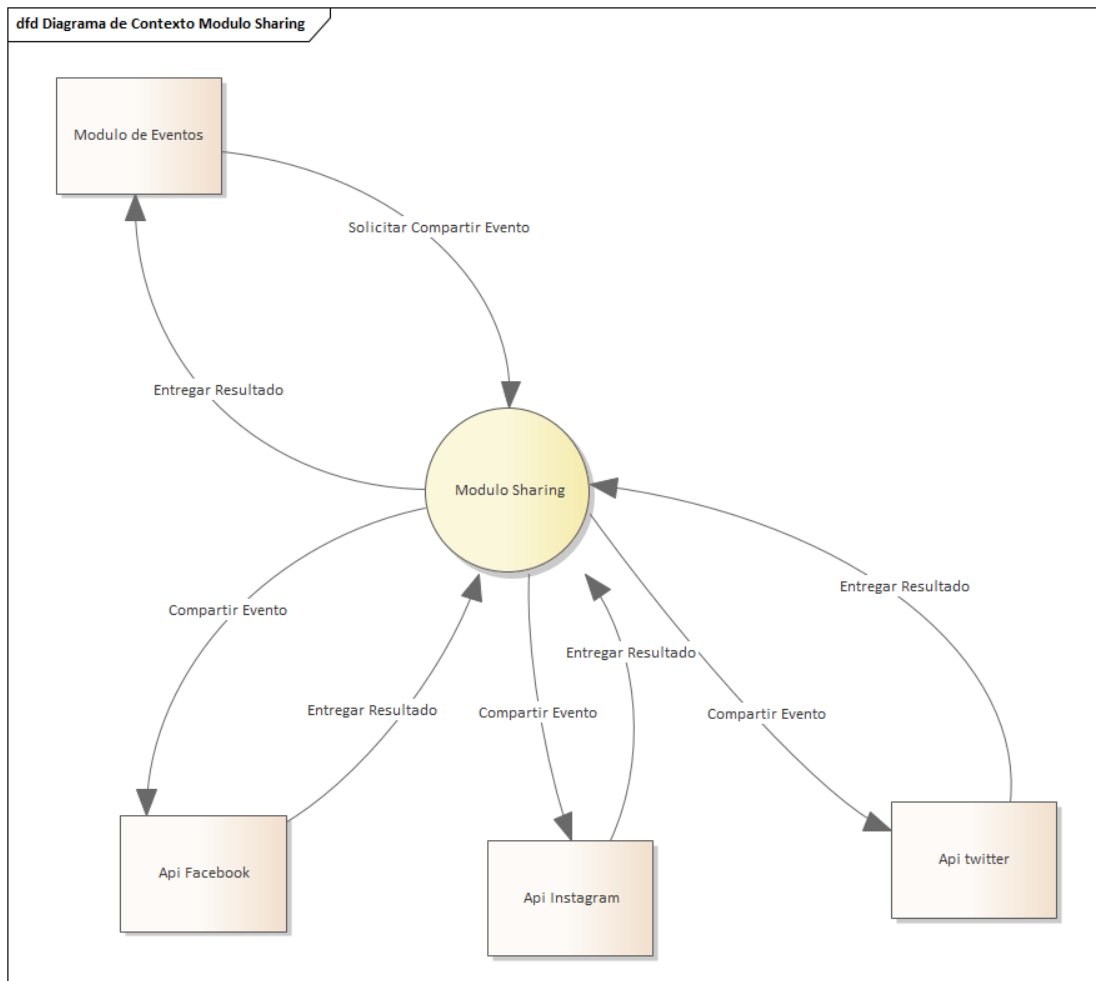


Figura 64: Diagrama de contexto del módulo de sharing

### Módulo Perfil

Se encarga de gestionar las operaciones que cada rol puede realizar en su perfil:

- Club:
  - Ver Perfil:
    - Imagen de perfil.
    - Nombre.
    - Email.
    - Teléfono.
    - Descripción.
    - Fotos del club.
    - Estado.
    - Calificación promedio.
  - Modificar su perfil:
    - Nombre.
    - Domicilio.
    - Cambiar imagen de perfil.

- Ver Canchas:
  - Nombre de la cancha.
  - Deporte de la cancha.
- Dar alta Cancha:
  - Seleccionar el deporte al que pertenece.
  - Indicar el nombre de la cancha.
- Dar baja Cancha.
- Modificar Cancha:
  - Cambiar el deporte al que pertenece.
  - Cambiar el nombre de la cancha.
- Ver equipos:
  - Ver los nombres de los equipos.
- Crear equipo:
  - Seleccionar deportistas que conformarán al equipo.
- Modificar equipo:
  - Agregar deportistas al equipo.
  - Eliminar deportistas del equipo.
- Eliminar equipo.
- Darse de baja.
- Deportista:
  - Ver Perfil:
    - Imagen de perfil.
    - Nombre.
    - Apellido.
    - Email.
    - Teléfono.
    - Descripción.
    - Estado.
  - Modificar su perfil:
    - Nombre.
    - Apellido.
    - DNI.
    - Fecha de nacimiento.
    - Cambiar imagen de perfil.
  - Crear equipo:
    - Seleccionar deportistas que conformarán al equipo.
  - Modificar equipo:
    - Agregar deportistas al equipo.
    - Eliminar deportistas del equipo.
  - Salir del equipo.
  - Darse de baja.
- Organizador
  - Ver Perfil:
    - Imagen de perfil.
    - Nombre.

- Email.
  - Teléfono.
  - Descripción.
  - Estado.
  - Calificación promedio.
  - Plan contratado.
  - Cantidad de eventos organizados.
- Modificar su perfil:
  - Nombre.
  - Teléfono.
  - CUIT.
  - Cambiar imagen de perfil.
- Darse de baja.
- Árbitro:
  - Modificar su perfil:
    - Nombre.
    - Apellido.
    - El deporte al que se dedica.
    - Cambiar imagen de perfil.
  - Darse de baja.
- Patrocinador:
  - Ver Perfil:
    - Imagen de perfil.
    - Nombre.
    - Email.
    - Teléfono.
    - Descripción.
    - Estado.
  - Modificar su perfil:
    - Nombre.
    - CUIT.
    - Teléfono.
    - Cambiar imagen de perfil.
  - Darse de baja.

## 2. Salidas del Sistema.

A continuación se enumeran las salidas del sistema:

### Módulo Seguridad

Número	1
Nombre	SignUp deportista
Descripción	Formulario para la gestión del registro del deportista en la plataforma.
Datos	<ul style="list-style-type: none"> <li>● Email: String</li> <li>● Nombre: String</li> <li>● Apellido: String</li> <li>● Teléfono: String</li> <li>● Localidad: String</li> <li>● Provincia: String</li> <li>● País: String</li> <li>● Contraseña: String</li> <li>● Imagen(opcional): Imagen</li> <li>● Fecha de nacimiento: Date</li> </ul>

Número	2
Nombre	SignUp club
Descripción	Formulario para la gestión del registro del club en la plataforma.
Datos	<ul style="list-style-type: none"> <li>● Razón social: String</li> <li>● CUIT: String</li> <li>● Domicilio social: String</li> <li>● Email: String</li> <li>● Contraseña: String</li> <li>● Teléfono: String</li> <li>● Imagen: Imagen</li> <li>● País: String</li> <li>● Provincia: String</li> <li>● Localidad: String</li> </ul>

Número	3
Nombre	SignUp patrocinador
Descripción	Formulario para la gestión del registro del patrocinador en la plataforma.

Datos	<ul style="list-style-type: none"> <li>● Razón social: String</li> <li>● CUIT: int</li> <li>● Domicilio social: String</li> <li>● Email: String</li> <li>● Contraseña: String</li> <li>● Teléfono: String</li> <li>● Imagen: Imagen</li> </ul>
-------	--

Número	4
Nombre	SignUp organizador
Descripción	Formulario para la gestión del registro del organizador en la plataforma.
Datos	<ul style="list-style-type: none"> <li>● Razón social: String</li> <li>● CUIT: String</li> <li>● Domicilio social: String</li> <li>● Email: String</li> <li>● Contraseña: String</li> <li>● Teléfono: String</li> <li>● Imagen: Imagen</li> </ul>

Número	5
Nombre	SignUp árbitro
Descripción	Formulario para la gestión del registro del árbitro en la plataforma.
Datos	<ul style="list-style-type: none"> <li>● Razón social: String</li> <li>● CUIT: String</li> <li>● Domicilio social: String</li> <li>● Email: String</li> <li>● Contraseña: String</li> <li>● Teléfono: String</li> <li>● Imagen: Imagen</li> </ul>

Número	6
Nombre	LogIn

Descripción	Permite al usuario ingresar a la aplicación
Datos	<ul style="list-style-type: none"> <li>• Email: String</li> <li>• Contraseña: String</li> </ul>

Número	7
Nombre	LogOut
Descripción	Permite al usuario cerrar sesión en la aplicación
Datos	-

Número	8
Nombre	Cambiar email
Descripción	Permite al usuario cambiar el mail
Datos	<ul style="list-style-type: none"> <li>• Email: String</li> <li>• Contraseña: String</li> <li>• Token: String</li> </ul>

Número	9
Nombre	Cambiar contraseña
Descripción	Permite al usuario cambiar la contraseña
Datos	<ul style="list-style-type: none"> <li>• Email: String</li> <li>• Nueva Contraseña: String</li> <li>• Contraseña: String</li> <li>• Token: String</li> </ul>

Número	10
--------	----



Nombre	Cambiar contraseña desde página principal
Descripción	Permite al usuario cambiar la contraseña
Datos	<ul style="list-style-type: none"> <li>● Email: String</li> <li>● Nueva Contraseña: String</li> </ul>

Número	11
Nombre	Contactarse con el administrador del sistema
Descripción	Permite al usuario contactar con los administradores del sistema.
Datos	<ul style="list-style-type: none"> <li>● Email: String</li> <li>● Nombre: String</li> <li>● Apellido: String</li> <li>● Teléfono: String</li> <li>● Campo de texto: String</li> </ul>

### Módulo Eventos

Número	12
Nombre	Alta evento
Descripción	Inmediatamente después de la creación del evento, se muestra a modo de resumen los aspectos más importantes del mismo.
Datos	<ul style="list-style-type: none"> <li>● Nombre Evento: String</li> <li>● Descripción evento: String</li> <li>● Imagen: Imagen</li> <li>● País: String</li> <li>● Provincia: String</li> <li>● Localidad: String</li> <li>● Club o dirección: String</li> <li>● Métodos de pago: String</li> <li>● Categorías: Object <ul style="list-style-type: none"> <li>○ Desde edad: int</li> <li>○ Hasta edad: int</li> <li>○ Nombre categoría: String</li> <li>○ Máx. Equipos categoría: int</li> </ul> </li> </ul>

	<ul style="list-style-type: none"> <li>○ Max partidos simultáneos: int</li> <li>○ Árbitros: List[Árbitros]</li> <li>○ Precio: float</li> <li>○ Jornadas <ul style="list-style-type: none"> <li>■ Horarios disponibles: DateTime</li> <li>■ Fecha: Date</li> </ul> </li> <li>○ Precio categoría: float</li> </ul>
--	--

Número	13
Nombre	Modificar evento
Descripción	Formulario para editar las características del evento, antes de que comience la inscripción para el mismo.
Datos	<ul style="list-style-type: none"> <li>● Descripción evento: String</li> <li>● Imagen: Imagen</li> <li>● País: String</li> <li>● Provincia: String</li> <li>● Localidad: String</li> <li>● Club o dirección: String</li> <li>● Métodos de pago: String</li> <li>● Categorías: Object <ul style="list-style-type: none"> <li>○ Desde edad: int</li> <li>○ Hasta edad: int</li> <li>○ Nombre categoría: String</li> <li>○ Máx. Equipos categoría: int</li> <li>○ Max partidos simultáneos: int</li> <li>○ Árbitros: List[Árbitros]</li> <li>○ Precio: float</li> <li>○ Jornadas <ul style="list-style-type: none"> <li>■ Horarios disponibles: DateTime</li> <li>■ Fecha: Date</li> </ul> </li> <li>○ Precio categoría: float</li> </ul> </li> </ul>

Número	14
Nombre	Eliminar Evento
Descripción	Permite "eliminar" un evento para cancelar

	la realización del mismo.
Datos	<ul style="list-style-type: none"> <li>• Nombre del evento: String</li> <li>• Nombre del estado del evento: String</li> </ul>

Número	15
Nombre	Listar peticiones de participación de evento como Club
Descripción	El usuario con rol del club desea listar todas las solicitudes pendientes de un evento.
Datos	<p>Para cada una de las solicitudes:</p> <ul style="list-style-type: none"> <li>• Descripción: String</li> <li>• Aceptar solicitud: Botón</li> <li>• Rechazar solicitud: Botón</li> </ul>

Número	16
Nombre	Ver eventos
Descripción	Un deportista desea visualizar todos los eventos de acuerdo a filtros específicos.
Datos	<p>Por cada evento:</p> <ul style="list-style-type: none"> <li>• Observaciones del evento: String</li> <li>• Fecha y hora del inicio del evento: DateTime</li> <li>• Fecha y hora de finalización del evento: DateTime</li> <li>• Precio de participación: float</li> <li>• Número máximo de equipos al evento: int</li> <li>• Número máximo de partidos simultáneos del evento: int</li> <li>• Patrocinadores del evento: List[Patrocinadores]</li> <li>• Estado del evento: String</li> <li>• Premios del evento por posición: float</li> <li>• Tipo del evento: String</li> </ul>

	<p>Filtros:</p> <ul style="list-style-type: none"> <li>● Por fecha y hora de inicio</li> <li>● Por fecha y hora de finalización</li> <li>● Por precio de participación</li> <li>● Por estado del evento</li> <li>● Por tipo de evento</li> </ul>
--	--

Número	17
Nombre	Visualizar evento aceptado en curso como árbitro
Descripción	Como usuario con rol Árbitro quiero visualizar los eventos que he aceptado y que se encuentran en curso para poder gestionarlos de forma rápida y fácil.
Datos	<p>Por cada uno de los eventos aceptados en curso:</p> <ul style="list-style-type: none"> <li>● Imagen del evento: Imagen</li> <li>● Descripción: String</li> <li>● Cantidad de personas inscriptas: int</li> <li>● Ubicación: String</li> <li>● Organizador: Organizador</li> <li>● Feedback: String</li> <li>● Equipos Inscriptos: int</li> <li>● Patrocinadores: Patrocinador</li> <li>● Estado: Estado</li> </ul>

Número	18
Nombre	Visualizar evento aceptado como árbitro
Descripción	Como usuario con rol Árbitro quiero visualizar los eventos que he aceptado para poder gestionarlos de manera rápida y fácil.
Datos	<p>Por cada uno de los eventos aceptados:</p> <ul style="list-style-type: none"> <li>● Imagen del evento: Imagen</li> <li>● Descripción: String</li> </ul>

	<ul style="list-style-type: none"> <li>• Cantidad de personas inscriptas: int</li> <li>• Ubicación: String</li> <li>• Organizador: Organizador</li> <li>• Feedback: String</li> <li>• Equipos Inscriptos: int</li> <li>• Patrocinadores: Patrocinador</li> <li>• Estado: Estado</li> </ul>
--	--

Número	19
Nombre	Visualizar eventos patrocinados como patrocinador
Descripción	Como usuario con rol de patrocinador quiero visualizar los eventos que he patrocinado para poder visualizarlos de manera rápida y fácil.
Datos	<p>Por cada uno de los eventos aceptados:</p> <ul style="list-style-type: none"> <li>• Imagen del evento: Imagen</li> <li>• Descripción: String</li> <li>• Cantidad de personas inscriptas: int</li> <li>• Ubicación: String</li> <li>• Organizador: Organizador</li> <li>• Feedback: String</li> <li>• Equipos Inscriptos: Int</li> <li>• Patrocinadores: Patrocinador</li> <li>• Estado: Estado</li> </ul>

Número	20
Nombre	Registrar resultado de partido
Descripción	Como usuario con roles de árbitro y organizador quiero registrar el resultado de un partido previamente asignado para poder identificar al ganador y perdedor.
Datos	<ul style="list-style-type: none"> <li>• Resultado equipo local: int</li> <li>• Resultado equipo visitante: int</li> </ul>

Número	21
Nombre	Ver resultado partido
Descripción	Permite visualizar los resultados de un partido, que pertenece a la jornada de un evento.
Datos	<ul style="list-style-type: none"> <li>• Nombre de los equipos: String</li> <li>• Nombre de los deportistas: String</li> <li>• Resultado del partido: String</li> <li>• Nombre del evento: String</li> <li>• Fecha y hora inicio partido: DateTime</li> <li>• Fecha y hora fin partido: DateTime</li> </ul>

Número	22
Nombre	Ver resultado de jornada
Descripción	Un usuario del sistema desea visualizar los resultados de la jornada de un evento específico.
Datos	<p>Por cada uno de los resultados de la jornada:</p> <ul style="list-style-type: none"> <li>• Nombre de los equipos: String</li> <li>• Nombre de los deportistas: String</li> <li>• Resultado de los eventos: int</li> <li>• Fecha y hora inicio eventos: DateTime</li> <li>• Fecha y hora fin eventos: DateTime</li> </ul>

Número	23
Nombre	Crear partido
Descripción	Permite la creación de un partido, para una jornada y evento en particular.
Datos	<ul style="list-style-type: none"> <li>• Nombre de los equipos: String</li> <li>• Nombre de los deportistas: String</li> <li>• Resultado del partido: int</li> <li>• Nombre del evento: String</li> <li>• Número de la jornada: int</li> <li>• Fecha y hora inicio partido:</li> </ul>

	<p>DateTime</p> <ul style="list-style-type: none"> <li>Fecha y hora fin partido: DateTime</li> </ul>
--	--

Número	24
Nombre	Listar árbitros
Descripción	Como usuario con rol Organizador quiero visualizar los árbitros disponibles para poder ver qué árbitros disponibles están cargados en el sistema
Datos	<p>Por cada uno de los árbitros:</p> <ul style="list-style-type: none"> <li>Nombre: String</li> <li>Apellido: String</li> </ul>

Número	25
Nombre	Alta jornada del evento
Descripción	Como organizador quiero crear jornadas de partidos para poder setear los equipos, horarios y canchas en las que se juegan.
Datos	<ul style="list-style-type: none"> <li>Cancha: String</li> <li>Fecha hora inicio: DateTime</li> <li>Fecha hora fin: DateTime</li> </ul>

Número	26
Nombre	Ver patrocinadores del evento
Descripción	Permite visualizar los distintos patrocinadores de un evento específico.
Datos	<p>Por cada patrocinador:</p> <ul style="list-style-type: none"> <li>Razón social: String</li> <li>Eventos realizados por el patrocinador: List[Evento]</li> <li>Imagen del patrocinador: String</li> </ul>

Módulo de inscripción

Número	27
Nombre	Inscribirse a evento como deportista
Descripción	Formulario que permite inscribirse a un evento.
Datos	<ul style="list-style-type: none"> <li>• Fecha y hora inscripción: DateTime</li> <li>• Observaciones: String</li> <li>• Estado inscripción: String</li> <li>• Nombre evento: String</li> <li>• Nombre deportista: String</li> <li>• Categoría de inscripción: String</li> <li>• Método de pago: String</li> <li>• Preferencia horaria: DateTime</li> </ul>

Número	28
Nombre	Pagar Inscripción
Descripción	El equipo paga la inscripción al evento deportivo.
Datos	<ul style="list-style-type: none"> <li>• Monto evento: float</li> <li>• Nombre equipo: String</li> <li>• Número de eventos: int</li> <li>• Nombre evento: String</li> <li>• Fecha transacción: DateTime</li> </ul>

Número	29
Nombre	Generar petición patrocinio como Patrocinador
Descripción	Como usuario con rol Patrocinador quiero generar una petición de patrocinio para poder patrocinar un evento.
Datos	<ul style="list-style-type: none"> <li>• Nombre evento: String</li> </ul>



Número	30
Nombre	Visualizar Peticiones de patrocinio como patrocinador
Descripción	Como usuario con rol Patrocinador quiero ver mis peticiones de patrocinio para poder cancelar o modificar una petición.
Datos	<ul style="list-style-type: none"> <li>• Nombre evento: String</li> <li>• Descripción del evento: String</li> <li>• Solicitud del evento: Solicitud</li> </ul>

Número	31
Nombre	Visualizar eventos patrocinados como patrocinador
Descripción	Como usuario con rol Patrocinador quiero observar mis eventos patrocinados para poder tener registro de los eventos que he patrocinado.
Datos	<ul style="list-style-type: none"> <li>• Nombre evento: String</li> <li>• Descripción del evento: String</li> <li>• Fecha de inicio del evento: DateTime</li> <li>• Fecha de finalización del evento: DateTime</li> </ul>

### Módulo de Feedback

Número	32
Nombre	Ver feedback de un evento
Descripción	Muestra todos los feedbacks del evento en particular.
Datos	<p>Por cada feedback:</p> <ul style="list-style-type: none"> <li>• Comentario feedback: String</li> <li>• Número feedback: int</li> <li>• Fecha emisión feedback: DateTime</li> <li>• Calificación: int</li> <li>• Nombre de usuario: String</li> </ul>

Número	33
Nombre	Realizar un feedback
Descripción	Formulario para que un usuario del sistema realice un feedback.
Datos	Nombre del deportista: String Nombre del evento: String Calificación del evento: int Observaciones del feedback:

### Módulo de sharing

Número	34
Nombre	Compartir evento
Descripción	El deportista desea obtener el link de un evento en particular para difundirlo en otras redes sociales y aumentar su difusión.
Datos	<ul style="list-style-type: none"> <li>• Link: String</li> </ul>

Número	35
Nombre	Compartir Equipo
Descripción	Como usuario con rol Deportista, Organizador o Patrocinador quiero compartir un Equipo para poder difundir el Equipo en las redes sociales y aumentar su popularidad.
Datos	<ul style="list-style-type: none"> <li>• Link: String</li> </ul>

Número	36
Nombre	Compartir perfil usuario
Descripción	Como usuario con Rol Organizador, Deportista, Patrocinador quiero compartir

	un perfil para poder difundirlo en las redes sociales
Datos	<ul style="list-style-type: none"> <li>• Link: String</li> </ul>

Número	37
Nombre	Vincular red social
Descripción	Como usuario quiero vincular mis redes sociales a mi perfil para poder ser encontrado en mis redes sociales por otros usuarios del sistema.
Datos	<ul style="list-style-type: none"> <li>• Link Facebook: String</li> <li>• Link Instagram: String</li> <li>• Link Twitter: String</li> <li>• Link LinkedIn: String</li> </ul>

#### Módulo de reportes

Número	38
Nombre	Ver cantidad de tipos de deportes usados para eventos
Descripción	Muestra gráficamente los tipos de deportes utilizados por el organizador en los eventos que él creó.
Datos	<ul style="list-style-type: none"> <li>• Por cada uno de los deportes una barra en el gráfico de barras indicando la cantidad de veces que este deporte es utilizado en un evento</li> </ul>

Número	39
Nombre	Ver cantidad de Inscriptos por mes
Descripción	Muestra la cantidad de deportistas inscriptos a los eventos que generó el organizador divididos por mes en un gráfico de barras.

Datos	<ul style="list-style-type: none"> <li>Cantidad de deportistas por mes.</li> <li>Nombre del organizador</li> </ul>
-------	--

Número	40
Nombre	Ver mejores equipos del evento
Descripción	Muestra los mejores equipos del evento.
Datos	<p>Por cada equipo que participó del evento:</p> <ul style="list-style-type: none"> <li>Nombre del equipo: String</li> <li>Imagen del equipo: String</li> <li>Puntos obtenidos: int</li> </ul>

Número	41
Nombre	Ver cantidad de eventos patrocinados
Descripción	Muestra la cantidad de eventos que ha realizado el organizador para saber la experiencia que tiene en la plataforma.
Datos	<ul style="list-style-type: none"> <li>Nombre del organizador: String</li> <li>Observaciones del organizador: String</li> <li>Número de eventos patrocinados: int</li> </ul>

Número	42
Nombre	Ver cantidad de tipos de organizadores del sistema
Descripción	Muestra gráficamente los tipos de organizadores del sistema.
Datos	<ul style="list-style-type: none"> <li>Por cada uno de los tipos de organizadores se muestra una barra en el gráfico de barras indicando la cantidad de veces que este tipo de organizador es utilizado.</li> </ul>

Módulo de Parametrización

Número	43
Nombre	Alta Tipo Deporte
Descripción	Formulario para dar de alta el tipo de deporte.
Datos	<ul style="list-style-type: none"> <li>• Nombre tipo deporte: String</li> <li>• Cantidad de equipos máximos: int</li> <li>• Cantidad de equipos mínimos: int</li> <li>• Cantidad de integrantes mínimos permitidos en un equipo: int</li> <li>• Cantidad de integrantes máximos permitidos en un equipo: int</li> <li>• Fecha inicio vigencia tipo deporte: DateTime</li> <li>• Fecha fin vigencia tipo deporte: DateTime</li> </ul>

Número	44
Nombre	Baja Tipo Deporte
Descripción	Formulario para dar de baja el tipo de deporte.
Datos	<ul style="list-style-type: none"> <li>• fecha fin vigencia tipo deporte: DateTime</li> </ul>

Número	45
Nombre	Modificar Tipo Deporte
Descripción	Formulario para modificar el tipo de deporte.
Datos	<ul style="list-style-type: none"> <li>• nombre tipo deporte: String</li> <li>• cantidad de equipos máximos: int</li> <li>• cantidad de equipos mínimos: int</li> <li>• cantidad de integrantes mínimos permitidos en un equipo: int</li> <li>• cantidad de integrantes máximos permitidos en un equipo: int</li> <li>• fecha inicio vigencia tipo deporte: DateTime</li> </ul>

Número	46
Nombre	Listar Tipos Deportes
Descripción	Muestra los distintos tipos de deportes que tiene el sistema
Historia de usuario relacionada	
Datos	<p>Por cada uno de los deportes:</p> <ul style="list-style-type: none"> <li>• Nombre del deporte: String</li> <li>• Máx. equipos por cancha: int</li> <li>• Min. equipos por cancha: int</li> <li>• Máx. integrantes por equipo: int</li> <li>• Min. integrantes por equipo: int</li> </ul>

Número	47
Nombre	Alta Tipo Organizador
Descripción	Formulario para dar de alta el tipo organizador.
Historia de usuario relacionada	40
Datos	<ul style="list-style-type: none"> <li>• Costo tipo organizador: double</li> <li>• Nombre tipo organizador: String</li> <li>• Valor agregado recomendación: float</li> <li>• Límite de evento: int</li> </ul>

Número	48
Nombre	Baja Tipo Organizador
Descripción	Formulario para dar de baja el tipo organizador.
Datos	<ul style="list-style-type: none"> <li>• Nombre tipo organizador: String</li> <li>• Fecha fin vigencia tipo organizador: DateTime</li> </ul>

Número	49
--------	----

Nombre	Modificar Tipo Organizador
Descripción	Formulario para modificar el tipo organizador.
Datos	<ul style="list-style-type: none"> <li>• Costo tipo organizador: double</li> <li>• Nombre tipo organizador: String</li> <li>• Valor agregado recomendación: float</li> <li>• Límite de evento: int</li> </ul>

Número	50
Nombre	Listar Tipos Organizadores
Descripción	Muestra los distintos tipos de organizadores del sistema.
Datos	<p>Por cada organizador:</p> <ul style="list-style-type: none"> <li>• Costo tipo organizador: double</li> <li>• Nombre tipo organizador: String</li> <li>• Valor agregado reputación: int</li> <li>• Límite eventos: int</li> <li>• Eventos realizados por mes: int</li> </ul>

Número	51
Nombre	Alta Rol
Descripción	Formulario para dar de alta un rol.
Datos	<ul style="list-style-type: none"> <li>• Nombre del rol: String</li> <li>• Descripción del rol: String</li> </ul>

Número	52
Nombre	Baja Rol
Descripción	Formulario para dar de baja un rol.
Datos	<ul style="list-style-type: none"> <li>• Nombre del rol: String</li> <li>• Descripción del rol: String</li> </ul>

Número	53
--------	----

Nombre	Modificar Rol
Descripción	Formulario para actualizar las características de un rol seleccionado.
Datos	<ul style="list-style-type: none"> <li>• Nombre del rol: String</li> <li>• Descripción del rol: String</li> <li>• Nombre permiso: String</li> <li>• Descripción permiso: String</li> </ul>

Número	54
Nombre	Listar Roles
Descripción	Muestra los distintos roles del sistema.
Datos	<p>Por cada rol:</p> <ul style="list-style-type: none"> <li>• Nombre del rol: String</li> <li>• Descripción del rol: String</li> <li>• Fecha fin vigencia rol: DateTime</li> <li>• Fecha inicio vigencia rol: String</li> </ul>

Número	55
Nombre	Alta Localidad
Descripción	Formulario para dar de alta una localidad.
Datos	<ul style="list-style-type: none"> <li>• Nombre localidad: String</li> </ul>

Número	56
Nombre	Baja Localidad
Descripción	Formulario para dar de baja una localidad.
Historia de usuario relacionada	
Datos	<ul style="list-style-type: none"> <li>• Nombre localidad: String</li> </ul>

Número	57
Nombre	Modificar Localidad
Descripción	Formulario para modificar un



	departamento.
Datos	<ul style="list-style-type: none"> <li>Nombre localidad: String</li> </ul>

Número	58
Nombre	Listar Localidades
Descripción	Muestras las localidades que tiene un departamento.
Datos	<p>Por cada una de las localidades:</p> <ul style="list-style-type: none"> <li>Nombre de localidad: String</li> <li>Nombre del departamento: String</li> </ul>

Número	59
Nombre	Alta Departamento
Descripción	Formulario para dar de alta un departamento.
Datos	<ul style="list-style-type: none"> <li>Nombre departamento: String</li> </ul>

Número	60
Nombre	Baja Departamento
Descripción	Formulario para dar de baja un departamento.
Datos	<ul style="list-style-type: none"> <li>Nombre departamento: String</li> <li>Fecha fin vigencia departamento: DateTime</li> </ul>

Número	61
Nombre	Modificar Departamento
Descripción	Formulario para modificar un

	departamento.
Datos	<ul style="list-style-type: none"> <li>Nombre departamento: String</li> </ul>

Número	62
Nombre	Listar Departamentos
Descripción	Muestras los departamentos que tiene una provincia.
Datos	<p>Por cada una de los departamentos:</p> <ul style="list-style-type: none"> <li>Nombre del departamento: String</li> <li>Nombre de la provincia: String</li> </ul>

Número	63
Nombre	Alta Tipo Evento
Descripción	Formulario para dar de alta el tipo evento.
Datos	<ul style="list-style-type: none"> <li>Nombre tipo deporte: String</li> <li>Descripción tipo deporte:String</li> <li>Cantidad de equipos máximos por cancha: int</li> <li>Cantidad de equipos mínimos por cancha: int</li> <li>Cantidad de integrantes mínimos permitidos en un equipo: int</li> <li>Cantidad de integrantes máximos permitidos en un equipo: int</li> <li>Fecha fin vigencia tipo deporte: DateTime</li> </ul>

Número	64
Nombre	Alta Provincia
Descripción	Formulario para dar de alta una provincia.
Datos	<ul style="list-style-type: none"> <li>nombre provincia: String</li> </ul>

Número	65
Nombre	Baja Provincia
Descripción	Formulario para dar de baja una provincia.
Datos	<ul style="list-style-type: none"> <li>nombre provincia: String</li> </ul>

Número	66
Nombre	Modificar Provincia
Descripción	Formulario para modificar una provincia.
Datos	<ul style="list-style-type: none"> <li>Nombre provincia: String</li> <li>Fecha fin vigencia provincia: DateTime</li> </ul>

Número	67
Nombre	Listar Provincias
Descripción	Muestra una lista de cada una de las provincias habilitadas por país
Datos	<p>Por cada uno de los países habilitados:</p> <ul style="list-style-type: none"> <li>Nombre de país: String</li> </ul> <p>Por cada una de las provincias de ese país:</p> <ul style="list-style-type: none"> <li>Nombre de la provincia: String</li> </ul>

Número	68
Nombre	Alta País
Descripción	El administrador visualiza la categoría

	creada en el sistema y el sistema muestra un mensaje de confirmación
Datos	<ul style="list-style-type: none"> <li>Nombre país: String</li> </ul>

Número	69
Nombre	Baja País
Descripción	Formulario para dar de baja un país.
Datos	<ul style="list-style-type: none"> <li>Nombre país: String</li> <li>Fecha fin vigencia país: DateTime</li> </ul>

Número	70
Nombre	Modificar País
Descripción	Formulario para modificar un país.
Datos	<ul style="list-style-type: none"> <li>Nombre país: String</li> </ul>

Número	71
Nombre	Listar Países
Descripción	Muestra una lista de cada uno de los países habilitados en la plataforma
Datos	Por cada uno de los países habilitados: <ul style="list-style-type: none"> <li>Nombre de país: String</li> </ul>

Número	72
Nombre	Baja Tipo Evento
Descripción	El usuario administrador desea dar de baja el tipo evento.
Datos	<ul style="list-style-type: none"> <li>Nombre tipo deporte: String</li> <li>Fecha fin vigencia tipo deporte: DateTime</li> </ul>

Número	73
--------	----

Nombre	Modificar Tipo Evento
Descripción	El usuario administrador visualiza el tipo de evento modificado en el sistema y el sistema muestra un mensaje de confirmación o error.
Datos	<ul style="list-style-type: none"> <li>• Nombre tipo deporte: String</li> <li>• Descripción tipo deporte: String</li> <li>• Cantidad de equipos máximos por cancha: int</li> <li>• Cantidad de equipos mínimos por cancha: int</li> <li>• Cantidad de integrantes mínimos permitidos en un equipo: int</li> <li>• Cantidad de integrantes máximos permitidos en un equipo: int</li> <li>• Fecha fin vigencia tipo deporte: DateTime</li> </ul>

Número	74
Nombre	Alta categoría
Descripción	El organizador quiere crear una categoría para poder agregar una nueva categoría y que pueda ser utilizado en los eventos
Datos	<ul style="list-style-type: none"> <li>• Nombre categoría: String</li> <li>• Observaciones: String</li> <li>• Edad mínima categoría: int</li> <li>• Edad máxima categoría: int</li> <li>• Cantidad mínima de integrantes por equipo: int</li> <li>• Cantidad máxima de integrantes por equipo</li> <li>• Deporte: Deporte</li> </ul>

Número	75
Nombre	Modificar categoría
Descripción	El organizador quiere crear una categoría para poder agregar una nueva categoría y que pueda ser utilizado en los eventos
Datos	<ul style="list-style-type: none"> <li>• Nombre categoría: String</li> </ul>

	<ul style="list-style-type: none"> <li>● Observaciones: String</li> <li>● Edad mínima categoría: int</li> <li>● Edad máxima categoría: int</li> <li>● Cantidad mínima de integrantes por equipo: int</li> <li>● Cantidad máxima de integrantes por equipo</li> <li>● Deporte: Deporte</li> </ul>
--	--

Número	76
Nombre	Listar categorías
Descripción	El organizador quiere visualizar todas las categorías creadas anteriormente
Datos	<p>Por cada una de las categorías habilitadas:</p> <ul style="list-style-type: none"> <li>● Nombre categoría: String</li> <li>● Observaciones: String</li> <li>● Edad mínima categoría: int</li> <li>● Edad máxima categoría: int</li> <li>● Cantidad mínima de integrantes por equipo: int</li> <li>● Cantidad máxima de integrantes por equipo</li> <li>● Deporte: Deporte</li> </ul>

### Módulo pagos

Número	77
Nombre	Realizar Pago en efectivo
Descripción	Registra un pago en efectivo para un deportista al inscribirse a un evento.
Datos	<ul style="list-style-type: none"> <li>● Nombre deportista: String</li> <li>● Nombre evento: String</li> <li>● Precio evento: int</li> <li>● Método de pago: String</li> <li>● Fecha fin inscripción: date</li> </ul>

Número	78
Nombre	Realizar Pago por mercado pago
Descripción	Registra un pago por mercado pago para un deportista al inscribirse a un evento.
Datos	<ul style="list-style-type: none"> <li>• Nombre deportista: String</li> <li>• Nombre evento: String</li> <li>• Precio evento: int</li> <li>• Método de pago: String</li> <li>• Fecha fin inscripción: date</li> </ul>

Número	79
Nombre	Confirmar Pago
Descripción	Confirma un pago como usuario organizador o árbitro para un deportista al pagar la inscripción a un evento.
Datos	<ul style="list-style-type: none"> <li>• Nombre deportista: String</li> <li>• Nombre evento: String</li> <li>• Precio evento: int</li> <li>• Estado Pago</li> <li>• Método de pago: String</li> <li>• Fecha fin inscripción: Date</li> </ul>

Número	80
Nombre	Cancelar Pago
Descripción	Cancela un pago como usuario organizador o árbitro para un deportista al no asistir a un evento o por decisión del actor.
Datos	<ul style="list-style-type: none"> <li>• Nombre deportista: String</li> <li>• Nombre evento: String</li> <li>• Precio evento: int</li> <li>• Estado Pago</li> <li>• Método de pago: String</li> <li>• Fecha fin inscripción: Date</li> </ul>

## Módulo Publicidad

Número	81
Nombre	Establecer Publicidad
Descripción	Como administrador quiero establecer la cantidad de publicidad que se muestra en cada página para poder ajustar el sitio a las necesidades del cliente
Datos	<ul style="list-style-type: none"> <li>• urlImagenSuperior: String</li> <li>• urlImagenInferior: String</li> <li>• Publicidad parte superior: Boolean</li> <li>• Publicidad parte inferior: Boolean</li> <li>• Imagen publicidad parte inferior: Imagen</li> <li>• Imagen publicidad parte superior: Imagen</li> </ul>

#### Módulo Perfil

Número	82
Nombre	Dar de alta cancha como club
Descripción	Crea una cancha perteneciente a un usuario tipo club
Datos	<ul style="list-style-type: none"> <li>• Nombre cancha: String</li> <li>• Capacidad cancha: int</li> <li>• Deportes cancha: Deportes</li> <li>• Número cancha: int</li> <li>• Precio por hora de cancha: int</li> </ul>

Número	83
Nombre	Modificar cancha como club
Descripción	Modifica una cancha perteneciente a un usuario tipo club
Datos	<ul style="list-style-type: none"> <li>• Nombre cancha: String</li> <li>• Capacidad cancha: int</li> <li>• Deportes cancha: Deportes</li> </ul>



	<ul style="list-style-type: none"> <li>• Número cancha: int</li> <li>• Precio por hora de cancha: int</li> </ul>
--	--

Número	84
Nombre	Dar de baja cancha como club
Descripción	Eliminar una cancha perteneciente a un usuario tipo club
Datos	<ul style="list-style-type: none"> <li>• Id cancha: String</li> </ul>

Número	85
Nombre	Visualizar canchas como club
Descripción	Un club desea visualizar todas las canchas que tiene a su disposición.
Datos	<p>Por cada una de las canchas:</p> <ul style="list-style-type: none"> <li>• Nombre cancha: String</li> <li>• capacidad cancha: int</li> <li>• deportes cancha: Deportes</li> <li>• Número cancha: int</li> <li>• precio por hora de cancha: int</li> </ul>

Número	86
Nombre	Visualizar perfil deportista
Descripción	Muestra en pantalla los datos de un perfil de tipo deportista a los demás usuarios del sistema.
Datos	<ul style="list-style-type: none"> <li>• Nombre usuario: String</li> <li>• Fecha Creación: fecha</li> <li>• Rol usuario: String</li> <li>• Teléfono usuario: String</li> <li>• Email Usuario: String</li> <li>• Redes Sociales: String</li> <li>• Deportes Favoritos: String</li> <li>• Eventos participados: Eventos</li> </ul>

Número	87
Nombre	Visualizar perfil organizador
Descripción	Muestra en pantalla los datos de un perfil de tipo organizador a los demás usuarios del sistema.
Datos	<ul style="list-style-type: none"> <li>● Nombre usuario: String</li> <li>● Fecha Creación: fecha</li> <li>● Rol usuario: String</li> <li>● Email Usuario: String</li> <li>● Teléfono usuario: String</li> <li>● Redes Sociales: String</li> <li>● Tipo de cuenta Organizador: String</li> <li>● Eventos Creados: Eventos</li> <li>● Feedback: int</li> </ul>

Número	88
Nombre	Visualizar perfil club
Descripción	Muestra en pantalla los datos de un perfil de tipo club a los demás usuarios del sistema.
Datos	<ul style="list-style-type: none"> <li>● Nombre usuario: String</li> <li>● Fecha Creación: fecha</li> <li>● Rol usuario: String</li> <li>● Email Usuario: String</li> <li>● Teléfono usuario: String</li> <li>● Redes Sociales: String</li> <li>● Tipo de cuenta Organizador: String</li> <li>● Eventos Creados: Eventos</li> <li>● Feedback: int</li> </ul>

Número	89
Nombre	Visualizar perfil árbitro
Descripción	Muestra en pantalla los datos de un perfil de tipo árbitro a los demás usuarios del sistema.

Datos	<ul style="list-style-type: none"> <li>• Nombre usuario: String</li> <li>• Fecha Creación: fecha</li> <li>• Rol usuario: String</li> <li>• Email Usuario: String</li> <li>• Teléfono usuario: String</li> <li>• Redes Sociales: String</li> <li>• Cantidad de eventos dirigidos: int</li> </ul>
-------	---

Número	90
Nombre	Visualizar perfil patrocinador
Descripción	Muestra en pantalla los datos de un perfil de tipo patrocinador a los demás usuarios del sistema.
Datos	<ul style="list-style-type: none"> <li>• Nombre usuario: String</li> <li>• Fecha Creación: fecha</li> <li>• Rol usuario: String</li> <li>• Email Usuario: String</li> <li>• Teléfono usuario: String</li> <li>• Redes Sociales: String</li> <li>• Cantidad de eventos participados: int</li> </ul>

Número	91
Nombre	Visualizar cancha
Descripción	Visualizar los datos de una cancha perteneciente a un usuario tipo club
Datos	<ul style="list-style-type: none"> <li>• Nombre cancha: String</li> <li>• Capacidad cancha: int</li> <li>• Deportes cancha: deportes</li> <li>• Número cancha: int</li> <li>• Precio por hora de cancha: int</li> </ul>

Número	92
Nombre	Listar equipos como deportista o club

Descripción	Visualizar los datos de los equipos a los que pertenece un usuario tipo club o deportista
Datos	<ul style="list-style-type: none"> <li>• Nombre equipo: String</li> <li>• Deportistas integrantes: String</li> <li>• Descripción: String</li> <li>• Club: String</li> </ul>

Número	93
Nombre	Crear equipos como deportista o club
Descripción	Crear un equipo como club o deportista
Datos	<ul style="list-style-type: none"> <li>• Nombre equipo: String</li> <li>• deportistas integrantes: String</li> <li>• descripción: String</li> <li>• Club: String</li> </ul>

Número	94
Nombre	Modificar equipos como deportista o club
Descripción	Modificar un equipo como club o deportista
Datos	<ul style="list-style-type: none"> <li>• Nombre equipo: String</li> <li>• Deportistas integrantes: String</li> <li>• Descripción: String</li> <li>• Club: String</li> </ul>

Número	95
Nombre	Dar de baja equipo como deportista o club
Descripción	Eliminar un equipo como club o deportista
Datos	<ul style="list-style-type: none"> <li>• ID equipo: String</li> </ul>

Número	96
Nombre	Salir de un equipo como deportista
Descripción	Salir de un equipo al que pertenece un

	usuario logueado como deportista
Datos	<ul style="list-style-type: none"> <li>ID equipo: String</li> </ul>

Número	97
Nombre	Modificar perfil como deportista
Descripción	Modificar los datos de un usuario logueado como deportista
Datos	<ul style="list-style-type: none"> <li>Nombre: String</li> <li>Apellido: String</li> <li>DNI:String</li> <li>FecaNacimiento: DateTime</li> <li>Teléfono: String</li> <li>foto de perfil: imagen</li> </ul>

Número	98
Nombre	Modificar perfil como club
Descripción	Modificar los datos de un usuario logueado como club
Datos	<ul style="list-style-type: none"> <li>Razón Social: String</li> <li>Localidad: String</li> <li>Provincia: String</li> <li>Departamento: String</li> <li>Dirección: String</li> <li>Teléfono: String</li> <li>Foto de perfil: imagen</li> </ul>

Número	99
Nombre	Modificar perfil como árbitro
Descripción	Modificar los datos de un usuario logueado como árbitro
Datos	<ul style="list-style-type: none"> <li>Nombre: String</li> <li>Apellido: String</li> <li>Teléfono: String</li> </ul>

	<ul style="list-style-type: none"> <li>• Foto de perfil: imagen</li> </ul>
--	--

Número	100
Nombre	Modificar perfil como patrocinador
Descripción	Modificar los datos de un usuario logueado como patrocinador
Datos	<ul style="list-style-type: none"> <li>• Nombre: String</li> <li>• CUIT: String</li> <li>• Teléfono: String</li> <li>• Foto de perfil: imagen</li> </ul>

Número	101
Nombre	Modificar perfil como organizador
Descripción	Modificar los datos de un usuario logueado como organizador
Datos	<ul style="list-style-type: none"> <li>• Nombre: String</li> <li>• CUIT: String</li> <li>• Teléfono: String</li> <li>• Foto de perfil: imagen</li> </ul>

Número	102
Nombre	Agregar deporte favorito como deportista
Descripción	Modificar los datos de un usuario logueado como deportista
Datos	<ul style="list-style-type: none"> <li>• Nombre: String</li> <li>• CUIT: String</li> <li>• Teléfono: String</li> <li>• Foto de perfil: imagen</li> </ul>

Número	103
Nombre	Quitar rol a usuario como administrador
Descripción	Eliminar un rol asociado a un usuario logeado como otro usuario administrador
Datos	<ul style="list-style-type: none"> <li>• Nombre rol: String</li> <li>• Nombre usuario: String</li> </ul>

Número	104
Nombre	Agregar rol a usuario como administrador
Descripción	Agregar un rol asociado a un usuario logeado como otro usuario administrador
Datos	<ul style="list-style-type: none"> <li>• Nombre rol: String</li> <li>• Nombre usuario: String</li> </ul>

Número	105
Nombre	Quitar rol a usuario propio
Descripción	Eliminar un rol asociado a un usuario logeado ese mismo usuario
Datos	<ul style="list-style-type: none"> <li>• Nombre rol: String</li> </ul>

Número	106
Nombre	Agregar rol a usuario propio
Descripción	Agregar un rol asociado a un usuario logeado ese mismo usuario
Datos	<ul style="list-style-type: none"> <li>• Nombre rol: String</li> </ul>

Número	107
--------	-----

Nombre	Dar de baja usuario propio
Descripción	Dar de baja el usuario propio del sistema
Datos	<ul style="list-style-type: none"> <li>Nombre usuario: String</li> </ul>

Número	108
Nombre	Crear equipo como club
Descripción	Como usuario con rol Club quiero crear un equipo para que el equipo pueda participar en un evento representando al club y dejar constancia de ello.
Datos	<ul style="list-style-type: none"> <li>Imagen del equipo: Imagen</li> <li>Nombre del equipo: String</li> <li>Descripción del equipo: String</li> <li>Integrantes: Array[Integrante]</li> </ul>

### 3. Modelo funcional

#### Historias de Usuario

Una historia de usuario es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final. Su propósito es articular cómo proporcionará una función de software valor al cliente.

Una historia de usuario es la unidad de trabajo más pequeña en un marco ágil. Es un objetivo final, no una función, expresado desde la perspectiva del usuario del software.

Una historia de usuario es una explicación general e informal de una función de software escrita desde la perspectiva del usuario final o cliente.

#### Formato Historia de Usuario

<b>Historia de usuario:</b> x
<b>Título:</b> Título de la historia de usuario



Descripción: <b>Como</b> (tipo/s de usuario que va a realizar la acción) <b>quiero</b> (realizar una acción en el sistema) <b>para poder</b> (intención de la acción).	
<b>Prioridad:</b> nivel de importancia de la hu	<b>Estimación:</b> horas que tomará el desarrollo de la hu
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
El usuario realice una acción	Respuesta que debe tener el sistema

Y para realizar los criterios de aceptación de cada user story utilizamos la metodología SMART, que indica que cada criterio de aceptación debe ser:

**Specific(específico):** Un objetivo específico se circunscribe a un aspecto, tarea o acción determinada de una empresa. Este objetivo es específico porque nos dice exactamente lo que esperamos lograr.

**Measurable(medible):** Para que sea mensurable, una meta tiene que ser específica. De otra manera, no es posible interpretar si los resultados están dentro de lo esperado. Además, es necesario contar con los medios para poder medirla, ya sea herramientas de software o una metodología de análisis que posibilite saber en qué medida se alcanzó el resultado previsto. Para medir cada uno de los criterios de aceptación, estos irán acompañados de sus test cases extraídos mediante la variante de caja negra "Partición Equivalente"

**Achievable(Alcanzable):** Por alcanzable nos referimos a un objetivo que sea perfectamente realizable en las condiciones con las que se cuenta. Es muy importante plantearse metas realistas, teniendo en cuenta que sobre la base de su cumplimiento se puede, luego, apuntar más alto. Por otro lado, en este punto es fundamental especificar a través de qué acción o acciones se pretende lograr dicho objetivo.

**Relevant(Relevante):** Una meta relevante es aquella que está en línea con los objetivos generales del negocio.

**Time-Boxed (Temporal):** Que los objetivos SMART sean temporales significa que están limitados a un tiempo determinado. Todas las características mencionadas su especificidad y mensurabilidad, su alcance y relevancia dependen del tiempo en que deben ser completadas.

## Especificación de historias de usuario

### Módulo Seguridad

Cada login específico posee una pantalla de registro con datos comunes a todos los usuarios.

<b>Número:</b> 1	
<b>Título:</b> SignUp Deportista	
<b>Descripción:</b> Como potencial usuario con rol deportista <b>quiero</b> crear una cuenta <b>para poder</b> identificarme en el sistema.	
<b>Pantalla:</b> Figura N.º 66	
<b>Prioridad:</b> 100	<b>Estimación:</b> 10
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Ingrese mi email para registrarse.	Que no permita un mail con caracteres especiales, ni un mail que ya esté previamente registrado en la plataforma.
Ingrese la contraseña para registrarme.	Que solo permita una contraseña que tenga al menos una mayúscula, una minúscula, un número y un carácter especial.
Ingrese mi confirmación de contraseña	Que sea la misma cadena de caracteres que ingrese en el campo "Contraseña"
Quiero guardar los datos ingresados.	Que se verifique que los campos: nombre, apellido, mail, contraseña.
Ingrese un nombre para registrarse	Que no permita el ingreso de nombres ofensivos u extravagantes.
Ingrese un teléfono para registrarse	Que no permita el ingreso de teléfonos inválidos, es decir deberá contener 13

	dígitos en total (contando el +54 9)
Quiero guardar los datos ingresados.	Que el sistema verifique que no existe ninguna otra cuenta con el mismo email ingresado.
Quiero guardar los datos ingresados.	Que el sistema verifique que no existe ningún link de confirmación que represente el email ingresado
Guardo los datos ingresados y estos son correctos.	Que el sistema envíe un link de confirmación al email ingresado
Acceda al link que está en mi email antes de que pase 1 día desde que cree la solicitud para crear una cuenta.	Que el sistema verifique la cuenta correctamente.
Acceda al link que está en mi email después de que pase 1 día desde que cree la solicitud para generar una cuenta.	Que el sistema informe al usuario que el link ha caducado

<b>Historia de usuario: 2</b>	
<b>Título:</b> Sign Up Club	
<b>Descripción:</b> Como potencial usuario con rol Club <b>quiero</b> crear una cuenta <b>para poder</b> identificarte en el sistema.	
<b>Pantalla:</b> Figura N.º 68	
<b>Prioridad:</b> 100	<b>Estimación:</b> 10
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Ingrese mi email para registrarse.	Que no permita un usuario con caracteres especiales.
Ingrese la contraseña para registrarme.	Que solo permita una contraseña que tenga al menos una mayúscula, una

	minúscula, un número y un carácter especial.
Ingrese mi domicilio	Que el sistema usando API de Google maps encuentre el domicilio en un mapa y extraiga la latitud y longitud del lugar seleccionado
Ingrese mi razón social	Que el sistema no me permita ingresar menos de 2 caracteres
Quiero guardar los datos ingresados.	Que se verifique que los campos: nombre, razón Social, teléfono, email, contraseña, confirmar contraseña, domicilio, localidad, departamento y provincia, no estén vacíos.
Quiero guardar los datos ingresados.	Que el sistema compruebe que no existe ninguna otra cuenta con el mismo email ingresado.
Quiero guardar los datos ingresados.	Que el sistema verifique que no existe ningún token de confirmación que represente el email ingresado
Guardo los datos ingresados y estos son correctos	Que el sistema envíe un link de confirmación al email ingresado
Acceda al link que está en mi email antes de que pase 1 día desde que cree la solicitud para crear una cuenta	Que el sistema verifique la cuenta correctamente.
Acceda al link que está en mi email después de que pase 1 día desde que cree la solicitud para crear una cuenta	Que el sistema informe al usuario que el link ha caducado

**Historia de usuario: 3**

**Título:** SignUp Patrocinador

**Descripción:** Como potencial usuario con rol Patrocinador **quiero** crear una cuenta **para poder** identificarme en el sistema.

**Pantalla:** Figura N.º 67

<b>Prioridad: 45</b>	<b>Estimación: 2</b>
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Ingrese mi email para registrarse.	Que no permita un usuario con caracteres especiales.
Ingrese la contraseña para registrarme.	Que solo permita una contraseña que tenga al menos una mayúscula, una minúscula, un número y un carácter especial.
Quiero guardar los datos ingresados.	Que se verifique que los campos: nombre de usuario, teléfono, email, contraseña, domicilio social, razón social y la CUIT, no estén vacíos.
Quiero guardar los datos ingresados.	Que el sistema revise que no existe ninguna otra cuenta con el mismo email ingresado.
Quiero guardar los datos ingresados.	Que el sistema verifique que no existe ningún token de confirmación que represente el email ingresado
Guardo los datos ingresados y estos son correctos	Que el sistema envíe un link de confirmación al email ingresado
Acceda al link que está en mi email antes de que pase 1 día desde que cree la solicitud para crear una cuenta	Que el sistema compruebe la cuenta correctamente.
Acceda al link que está en mi email después de que pase 1 día desde que cree la solicitud para crear una cuenta	Que el sistema informe al usuario que el link ha caducado

<b>Historia de usuario: 4</b>
<b>Título: SignUp Organizador</b>

**Descripción:** Como potencial usuario con rol Organizador **quiero** crear una cuenta **para poder** identificarme en el sistema.

**Pantalla:** Figura N.º 69

**Prioridad:** 100

**Estimación:** 4

**Criterios de aceptación**

**Cuando**

**Espero**

Ingrese mi email para registrarse.

Que no permita un usuario con caracteres especiales.

Ingrese la contraseña para registrarme.

Que solo permita una contraseña que tenga al menos una mayúscula, una minúscula, un número y un carácter especial.

Quiero guardar los datos ingresados.

Que se verifique que los campos: nombre, nombre de usuario, teléfono, email, contraseña, CUIT, tipo de organizador y la provincia en la que operará, no estén vacíos.

Quiero guardar los datos ingresados.

Que el sistema verifique que no existe ninguna otra cuenta con el mismo email ingresado.

Quiero guardar los datos ingresados.

Que el sistema verifique que no existe ningún token de confirmación que represente el email ingresado

Guardo los datos ingresados y estos son correctos

Que el sistema envíe un link de confirmación al email ingresado

Acceda al link que está en mi email antes de que pase 1 día desde que cree la solicitud para crear una cuenta

Que el sistema verifique la cuenta correctamente.

Acceda al link que está en mi email después de que pase 1 día desde que cree la solicitud para crear una cuenta

Que el sistema informe al usuario que el link ha caducado

<b>Historia de usuario: 5</b>	
<b>Título:</b> SignUp Arbitro	
<b>Descripción:</b> Como potencial usuario con rol Árbitro <b>quiero</b> crear una cuenta <b>para poder</b> identificarme en el sistema.	
<b>Pantalla:</b> Figura N.º 70	
<b>Prioridad:</b> 45	<b>Estimación:</b> 2
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Ingrese mi email para registrarse.	Que no permita un usuario con caracteres especiales.
Ingrese la contraseña para registrarme.	Que solo permita una contraseña que tenga al menos una mayúscula, una minúscula, un número y un carácter especial.
Quiero guardar los datos ingresados.	Que se verifique que los campos: nombre, teléfono, email, contraseña, DNI y la provincia en donde operará no estén vacíos.
Quiero guardar los datos ingresados.	Que el sistema revise que no existe ninguna otra cuenta con el mismo email ingresado.
Quiero guardar los datos ingresados.	Que el sistema verifique que no existe ningún token de confirmación que represente el email ingresado
Guardo los datos ingresados y estos son correctos	Que el sistema envíe un link de confirmación al email ingresado
Acceda al link que está en mi email antes de que pase 1 día desde que cree la solicitud para crear una cuenta	Que el sistema verifique la cuenta correctamente.
Acceda al link que está en mi email después de que pase 1 día desde que	Que el sistema informe al usuario que el link ha caducado

crea la solicitud para crear una cuenta	
---	--

<b>Historia de usuario: 6</b>	
<b>Título:</b> LogIn	
<b>Descripción:</b> Como usuario <b>quiero</b> identificarme en el sistema con mi email y contraseña <b>para poder</b> hacer uso de las funcionalidades del sistema.	
<b>Pantalla:</b> Figura N.º 71	
<b>Prioridad:</b> 100	<b>Estimación:</b> 25
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Ingrese mi email.	Que no permita caracteres especiales.
Quiero ingresar al sistema y el email o la contraseña ingresadas no sean correctas	Que no se me permita el acceso y se me informe que los datos ingresados son incorrectos y por qué son incorrectos
Quiero ingresar al sistema y el email y la contraseña ingresadas sean correctas	Que me permita acceder al sistema mostrándome una pantalla temporal de usuario logueado con éxito, y luego me redirija a la página principal del sistema.
Quiero ingresar al sistema y mi rol fue dado de baja	Que el sistema me muestra una lista de tipos de roles básicos, que al seleccionar uno se me asigne y se me redirija a la pantalla principal del sistema
Quiero ingresar al sistema y la última vez que me loguee al sistema marque la opción de recordarme	Que el sistema me autocomplete los datos de email y contraseña
Ha pasado más de 2 horas	Qué mi sesión expire



<b>Historia de usuario: 7</b>	
<b>Título</b> Log Out	
<b>Descripción:</b> yo <b>como</b> usuario de la aplicación <b>quiero</b> cerrar sesión <b>para poder</b> salir de la misma correctamente.	
<b>Pantalla:</b> -	
<b>Prioridad:</b> 20	<b>Estimación:</b> 3
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Estoy logueado como usuario y hago clic en el botón cerrar sesión	El sistema termina mi sesión en el navegador y borra mi token de sesión de los tokens activos en la base de datos.

<b>Historia de usuario: 8</b>	
<b>Título</b> Cambiar Email	
<b>Descripción:</b> yo <b>como</b> usuario de la aplicación <b>quiero</b> cambiar el mail	
<b>Pantalla:</b> Figura N.º 142	
<b>Prioridad:</b> 20	<b>Estimación:</b> 8
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Ingrese el nuevo email.	Que no permita un mail con caracteres

	especiales, ni un mail que ya esté previamente registrado en la plataforma.
Ingrese contraseña errónea y clickee en guardar	Que el sistema me muestre un mensaje de: "contraseña errónea, por favor ingrese su contraseña"
Ingrese el nuevo email y la contraseña correcta y clickee en guardar	Que el sistema guarde el mail ingresado como mail actual del usuario y me envíe al mail avisando el cambio de contraseña

<b>Historia de usuario:</b> 9	
<b>Título</b> Cambiar Contraseña	
<b>Descripción:</b> yo <b>como</b> usuario de la aplicación <b>quiero</b> cambiar la contraseña desde mi perfil <b>para poder</b> aumentar la seguridad y privacidad del usuario.	
<b>Pantalla:</b> Figura N.º 143	
<b>Prioridad:</b> 30	<b>Estimación:</b> 7
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Ingrese la nueva contraseña.	Que solo permita una contraseña que tenga al menos una mayúscula, una minúscula, un número y un carácter especial.
Ingrese mi confirmación de contraseña	Que sea la misma cadena de caracteres que ingrese en el campo "Contraseña"
Ingrese la nueva contraseña.	Que la contraseña no sea la misma que la actual ni la anterior.
Clickee en cambiar contraseña y mi contraseña actual sea la misma que la ingresada en el campo de contraseña actual	Que el sistema cambie mi contraseña actual por la contraseña nueva ingresada

Clickee en cambiar contraseña y mi contraseña actual no sea la misma que la ingresada en el campo de contraseña actual por primera o segunda vez	Que el sistema no cambie mi contraseña actual por la contraseña nueva ingresada, me muestre un cartel de “la contraseña actual ingresada no es correcta” y me permita cargar los datos nuevamente
Clickee en cambiar contraseña y mi contraseña actual no sea la misma que la ingresada en el campo de contraseña actual por tercera vez	Que el sistema no cambie mi contraseña actual por la contraseña nueva ingresada, me muestre un cartel de “Ha intentado demasiadas veces cambiar la contraseña, su cuenta será bloqueada por favor chequee su mail para reactivarla”, me envíe un mail para reactivar mi cuenta y me desloguee del sistema

<b>Historia de usuario:</b> 10	
<b>Título</b> Cambiar Contraseña desde email	
<b>Descripción:</b> yo <b>como</b> usuario de la aplicación <b>quiero</b> recuperar la contraseña desde la página de logueo <b>para poder</b> loguearme nuevamente al sistema o aumentar la seguridad de mi cuenta.	
<b>Pantalla:</b> Figura N.º 232, 233	
<b>Prioridad:</b> 75	<b>Estimación:</b> 10
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Ingrese el mail	Que solo me permita ingresar texto con formato de mail
Ingrese un mail no asociado a un usuario	Que me muestre un cartel con “ese mail no corresponde a ningún usuario del sistema” y me permita luego ingresar mi mail
Ingrese un mail asociado a un usuario existente y habilitado	Que me envíe a ese email un enlace para acceder a mi cuenta y cambiar mi

	contraseña
Ingreso al enlace enviado a mi mail	Que me habilite 2 campos 1 para ingresar la nueva contraseña y otro para confirmar la nueva contraseña
Ingrese la nueva contraseña.	Que solo permita una contraseña que tenga al menos una mayúscula, una minúscula, un número y un carácter especial.
Ingrese mi confirmación de contraseña	Que sea la misma cadena de caracteres que ingrese en el campo "Contraseña"
Ingrese la nueva contraseña.	Que la contraseña no sea la misma que la actual ni la anterior.
Ingreso una nueva contraseña correcta, con su adecuada confirmación y clickeo en cambiar contraseña	Que el sistema cambie mi contraseña actual por la contraseña nueva ingresada

<b>Historia de usuario: 11</b>	
<b>Título:</b> Contactarse con el administrador del sistema	
<b>Descripción:</b> Como usuario <b>quiero poder</b> contactarme con el administrador del sistema <b>para poder</b> informarle sobre errores que se han producido al usar la aplicación o pedirle nuevas funcionalidades.	
<b>Pantalla:</b> Figura N.º 101	
<b>Prioridad:</b> 5	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Completo el campo con el mensaje que quiero enviarle a los administradores, los demás campos de nombre, apellido, mail, y teléfono y clickear en el botón enviar.	Que el sistema envíe al mail del administrador el mensaje especificado por el usuario y me muestre un mensaje de éxito de envío

Alguno de los campos obligatorios está vacío	Que el sistema me muestre un mensaje de "Por favor complete los campos correspondientes obligatorios"
--	---

<b>Historia de usuario: 12</b>	
<b>Título:</b> Realizar backup base de datos	
<b>Descripción:</b> Como usuario con rol Administrador <b>quiero poder</b> realizar copias de seguridad de la base de datos <b>para poder</b> tener una copia de seguridad de la base de datos y poder restaurarla según se necesite	
<b>Pantalla:</b> Figuras N.º 352,353	
<b>Prioridad:</b> 50	<b>Estimación:</b> 10
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Me dirijo al menú de backups en el sidebar lateral izquierdo en el menú del administrador	Ver una pantalla con 2 opciones: "Realizar backup" y "Restaurar backup"
Hago click en el icono de Realizar Backup	Espero descarga un archivo comprimido tar.gz con la base de datos completa del sistema

<b>Historia de usuario: 13</b>	
<b>Título:</b> Restaurar base de datos de un backup previo	
<b>Descripción:</b> Como usuario con rol Administrador <b>quiero poder</b> restaurar la base de datos del sistema <b>para poder</b> volver todos los datos del sistema a un estado conocido	

y seguro en caso de emergencia	
<b>Pantalla:</b> Figuras N.º 50	
<b>Prioridad:</b> 50	<b>Estimación:</b> 10
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Me dirijo al menú de backups en el sidebar lateral izquierdo en el menú del administrador	Ver una pantalla con 2 opciones: “Realizar backup” y “Restaurar backup”
Cuando hago click en el icono de restaurar la base de datos	Que el sistema me permita seleccionar un archivo desde mi computadora y al cargarlo en el sistema que los datos de la base de datos se restauren al punto seleccionado.

Módulo de eventos

<b>Historia de usuario:</b> 14	
<b>Título:</b> Alta Evento	
<b>Descripción:</b> Como usuario con rol Organizador <b>quiero</b> crear un evento <b>para poder</b> registrar un nuevo evento bajo mi nombre para que los deportistas se puedan inscribir.	
<b>Pantalla:</b> Figura N° 73,74	
<b>Prioridad:</b> 100	<b>Estimación:</b> 50
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Indique que el tipo de evento es un torneo.	Que me permita enviar peticiones a un solo club.
Tenga dos categorías con el mismo nombre.	Que el sistema no me permita ingresar la segunda categoría con el mismo nombre
Ingrese el precio de inscripción de la categoría.	Que el sistema solo me permita ingresar números positivos.
Ingrese la cantidad máxima de equipos en la categoría.	Que el sistema solo me permita ingresar números positivos.
Ingrese la fecha de inicio de inscripción antes de la fecha actual	Que el sistema no me permita guardar el evento.
Quiero guardar los datos ingresados.	Que se verifique que los campos: nombre Evento, categorías (nombre Categoría, días y horarios de juego, precio, cantidad máxima de equipos, cantidad mínima de jugadores equipo, cantidad máxima de jugadores equipo) y el tipo de evento no estén vacíos.
Quiero guardar el evento y los datos ingresados son correctos.	Que el sistema guarde el evento con estado "Propuesto" y que el sistema

	notifique al/los clubes seleccionados.
--	--

<b>Historia de usuario: 15</b>	
<b>Título:</b> Modificar Evento	
<b>Descripción:</b> Como usuario con rol Organizador <b>quiero</b> modificar un evento <b>para poder</b> actualizar las características del mismo y que los usuarios del sistema puedan ver el evento modificado.	
<b>Pantalla:</b> Figura N° 75	
<b>Prioridad:</b> 90	<b>Estimación:</b> 40
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Tenga dos categorías con el mismo nombre.	Que el sistema no me permita guardar los cambios.
Ingrese el precio de inscripción de la categoría.	Que el sistema solo me permita ingresar números.
Quiero guardar los datos ingresados.	Que se verifique que los campos: nombre Evento, categorías (nombre Categoría, días y horarios de juego, precio, cantidad máxima de equipos, cantidad mínima de jugadores equipo, cantidad máxima de jugadores equipo) y el tipo de evento no estén vacíos.
Quiero guardar el evento y la fecha actual sea mayor o igual a la fecha inicio inscripción	Que el sistema no guarde los cambios realizados sobre el evento y notifique al usuario que no es posible modificar el evento
Quiero guardar el evento y los datos ingresados son correctos.	Que el sistema guarde los cambios realizados sobre el evento



<b>Historia de usuario: 16</b>	
<b>Título:</b> Eliminar Evento	
<b>Descripción:</b> Como usuario con rol Organizador <b>quiero</b> eliminar un evento <b>para poder</b> hacer que los usuarios del sistema ya no lo vean como un evento a realizarse.	
<b>Pantalla:</b> Figuras N.º 76	
<b>Prioridad:</b> 70	<b>Estimación:</b> 15
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera eliminar el evento y la fecha actual y ya haya pasado la fecha fin inscripción	Que el sistema no me permita eliminar el evento
Quiera eliminar el evento, la fecha actual y no haya pasado la fecha fin inscripción y el estado del evento es "Propuesto"	Que el sistema modifique al evento y lo relacione al estado "Cancelado"
Quiera eliminar el evento, y la fecha actual no haya pasado la fecha fin inscripción y el estado del evento es "Inscripciones en curso"	Que el sistema modifique al evento y lo relacione al estado "Cancelado" y que notifique a todos los clubes, deportistas y patrocinadores relacionados con el evento

<b>Historia de usuario: 17</b>
<b>Título:</b> Listar peticiones de participación de evento como Club
<b>Descripción:</b> Como usuario con rol Club <b>quiero</b> las peticiones de eventos pendientes <b>para poder</b> aceptar o rechazarlas de forma rápida.

<b>Pantalla:</b> Figura N.º 82	
<b>Prioridad:</b> 100	<b>Estimación:</b> 40
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Presiono ver peticiones de participación de eventos pendientes.	El sistema me muestre las peticiones pendientes de una forma fácil y rápida de observar.

<b>Historia de usuario:</b> 18	
<b>Título:</b> Aceptar petición evento como Club	
<b>Descripción:</b> Como usuario con rol Club <b>quiero</b> aceptar la petición de evento <b>para poder</b> llevar a cabo el evento en mi club.	
<b>Pantalla:</b> Figura N.º 83	
<b>Prioridad:</b> 100	<b>Estimación:</b> 40
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera aceptar un evento en mi club y sea el único club en donde se registró el evento	Que el sistema cambie el estado de EventoClub a "Aceptado", el estado del evento a "Confirmada" y notifique al organizador.
Quiera aceptar un evento en mi club, no sea el único club en donde se registró el evento y los otros clubes ya aceptaron la	Que el sistema cambie el estado de EventoClub a "Aceptado" y el estado del evento a "Confirmada"

petición	
Quiera aceptar un evento en mi club, no sea el único club en donde se registró el evento y al menos uno de los otros clubes todavía no acepta la petición.	Que el sistema cambie el estado de EventoClub a "Aceptado"

<b>Historia de usuario: 19</b>	
<b>Título:</b> Rechazar petición evento como Club	
<b>Descripción:</b> Como usuario con rol Club <b>quiero</b> rechazar la petición de evento <b>para poder</b> desvincular al club del evento.	
<b>Pantalla:</b> Figura N° 83	
<b>Prioridad:</b> 70	<b>Estimación:</b> 5
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera rechazar un evento en mi club	Que el sistema cambie el estado de Categoría Evento Club a "Rechazado", el estado del evento a "Cancelado" y notifique al organizador que su evento no fue aprobado

<b>Historia de usuario: 20</b>	
<b>Título:</b> Aceptar petición de patrocinio como Organizador	
<b>Descripción:</b> Como usuario con rol Organizador <b>quiero</b> aceptar la petición de patrocinio <b>para poder</b> vincular al patrocinador con el evento.	

<b>Pantalla:</b> Figura N.º 85	
<b>Prioridad:</b> 45	<b>Estimación:</b> 10
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Seleccione aceptar una petición de patrocinio en las notificaciones de mi perfil.	Que el sistema me muestre un cartel preguntándome si deseo aceptar realmente la petición de patrocinio.
Quiera seleccione confirmar aceptar la petición de patrocinio	Que el sistema guarde la petición de patrocinio con el estado "aceptado" y envíe una notificación por mail al organizador del evento asociando el patrocinador al evento . Y me muestre un cartel con un mensaje de éxito.

<b>Historia de usuario:</b> 21	
<b>Título:</b> Rechazar petición de patrocinio como Organizador	
<b>Descripción:</b> Como usuario con rol Organizador <b>quiero</b> rechazar la petición de patrocinio <b>para poder</b> impedir la vinculación de un patrocinador con un evento	
<b>Pantalla:</b> Figura N.º 85	
<b>Prioridad:</b> 45	<b>Estimación:</b> 5
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Seleccione rechazar una petición de patrocinio en las notificaciones de mi perfil.	Que el sistema me muestre un cartel preguntándome si deseo rechazar realmente la petición de patrocinio.

Quiera seleccione confirmar rechazar la petición de patrocinio	Que el sistema guarde la petición de patrocinio con el estado "rechazado" y envíe una notificación por mail al patrocinador del evento. Y me muestre un cartel con un mensaje de rechazo con éxito.
--	---

<b>Historia de usuario: 22</b>	
<b>Título:</b> Visualizar evento	
Descripción: <b>Como</b> usuario con rol Organizador, Deportista, Patrocinador <b>quiero</b> visualizar un evento <b>para</b> poder ver los datos de la misma en detalle.	
<b>Pantalla:</b> Figura N.º 90	
<b>Prioridad:</b> 100	<b>Estimación:</b> 40
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
En el listado de eventos seleccioné un evento deseado	Poder ver toda la información relacionada con el evento de forma detallada: <ul style="list-style-type: none"> <li>● Imagen del evento</li> <li>● Descripción</li> <li>● Cantidad de personas inscriptas</li> <li>● Ubicación</li> <li>● Organizador</li> <li>● Feedback</li> <li>● Equipos Inscriptos</li> <li>● Patrocinadores</li> <li>● Estado</li> </ul>
En el listado de eventos seleccioné un evento deseado	La información mostrada por el sistema sea confiable y corresponda al evento seleccionado
En el listado de eventos seleccioné un evento deseado	La información esté desplegada en una pantalla fácil de leer y la información esté

	distribuida de una forma clara
--	--------------------------------

<b>Historia de usuario: 23</b>	
<b>Título:</b> Visualizar evento aceptado en curso como árbitro	
<b>Descripción:</b> Como usuario con rol Árbitro <b>quiero</b> visualizar los eventos que he aceptado y que se encuentran en curso <b>para</b> poder gestionarlos de forma rápida y fácil.	
<b>Pantalla:</b> Figura N.º 90	
<b>Prioridad:</b> 20	<b>Estimación:</b> 20
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Presiono ver eventos aceptados en curso	<p>Poder ver toda la información relacionada con el evento de forma detallada:</p> <ul style="list-style-type: none"> <li>● Imagen del evento</li> <li>● Descripción</li> <li>● Cantidad de personas inscriptas</li> <li>● Ubicación</li> <li>● Organizador</li> <li>● Feedback</li> <li>● Equipos Inscriptos</li> <li>● Patrocinadores</li> <li>● Estado</li> </ul> <p>Y al presionar ver más información que se me redirija al apartado para poder gestionarlo.</p>

<b>Historia de usuario: 24</b>
<b>Título:</b> Visualizar evento aceptado como árbitro

**Descripción:** Como usuario con rol Árbitro **quiero** visualizar los eventos que he aceptado **para** poder gestionarlos de forma rápida y fácil.

**Pantalla:** Figura N.º 90

**Prioridad:** 20

**Estimación:** 20

**Criterios de aceptación**

**Cuando**

Presiono ver eventos aceptados

**Espero**

Poder ver toda la información relacionada con el evento de forma detallada:

- Imagen del evento
- Descripción
- Cantidad de personas inscriptas
- Ubicación
- Organizador
- Feedback
- Equipos Inscriptos
- Patrocinadores
- Estado

**Historia de usuario:** 25

**Título:** Visualizar eventos patrocinados como patrocinador

**Descripción:** Como usuario con rol de patrocinador **quiero** visualizar los eventos que he patrocinado **para** poder visualizarlos de manera rápida y fácil.

**Pantalla:** Figura N.º 173

**Prioridad:** 10

**Estimación:** 20

Criterios de aceptación	
Cuando	Espero
Presiono ver eventos patrocinados	Poder ver toda la información relacionada con el evento de forma detallada: <ul style="list-style-type: none"> <li>● Imagen del evento</li> <li>● Descripción</li> <li>● Cantidad de personas inscriptas</li> <li>● Ubicación</li> <li>● Organizador</li> <li>● Feedback</li> <li>● Equipos Inscriptos</li> <li>● Patrocinadores</li> <li>● Estado</li> </ul>

<b>Historia de usuario: 26</b>	
<b>Título:</b> Visualizar solicitudes de patrocinio pendientes como patrocinador	
<b>Descripción:</b> Como usuario con rol de patrocinador <b>quiero</b> visualizar mis solicitudes de patrocinio que todavía no han sido aceptadas <b>para</b> poder gestionarlas de forma rápida y fácil.	
<b>Pantalla:</b> Figura N.º 175	
<b>Prioridad:</b> 10	<b>Estimación:</b> 20
Criterios de aceptación	
Cuando	Espero
Presiono ver solicitudes de patrocinio pendientes.	El sistema me muestra información sobre mi petición de patrocinio y al evento al que fue realizada.



**Historia de usuario: 27**

**Título:** Cancelar solicitudes de patrocinio pendientes como patrocinador

**Descripción:** Como usuario con rol de patrocinador **quiero** cancelar una solicitud de patrocinio que todavía no ha sido aceptada **para** poder olvidarme de la misma y poder patrocinar otro evento.

**Pantalla:** -

**Prioridad:** 10

**Estimación:** 30

**Criterios de aceptación**

**Cuando**

Presiono ver solicitudes de patrocinio pendientes.

**Espero**

El sistema me muestra información sobre mi petición de patrocinio y al evento al que fue efectuada.

Presiono cancelar petición de patrocinio

El sistema cancela la petición de patrocinio y no permite que el organizador la acepte.

**Historia de usuario: 28**

**Título:** Registrar resultado de partido

**Descripción:** Como usuario con roles de árbitro y organizador **quiero** registrar el resultado de un partido previamente asignado **para poder** identificar al ganador y perdedor.

**Pantalla:** Figura N.º 175

<b>Prioridad:</b> 100	<b>Estimación:</b> 10
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Ingrese a la pestaña registrar resultados	Que el sistema muestre una lista con todos los eventos los cuales el árbitro tiene asignado uno o más partidos.
Ingrese al evento asignado.	Que solo me muestre los partidos asignados al árbitro y que no tiene ningún tipo de resultado todavía.
Ingrese un resultado para el equipo local.	Que solo permita caracteres numéricos.
Ingrese un resultado para el equipo visitante.	Que solo permita caracteres numéricos.
Guardo los datos ingresados y estos son correctos	Que el sistema determine al ganador y perdedor del sistema, y actualice las llaves del torneo.

<b>Historia de usuario:</b> 29	
<b>Título</b> Visualizar resultado de la jornada.	
<b>Descripción:</b> Como usuario <b>quiero</b> poder visualizar los resultados de la jornada de un evento específico <b>para poder</b> mantenerme informado del evento.	
<b>Pantalla:</b> Figura N.º 103	
<b>Prioridad:</b> 60	<b>Estimación:</b> 30
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Seleccione el Botón del resultado de una	El sistema me muestre cada uno de los

jornada dentro de un evento en específico	partidos acontecidos y a acontecer en la jornada con su respectiva información más relevante, ordenados por hora, vínculos a la información específica de cada partido, y equipos participantes listados de la jornada del evento y un botón de volver
Cuando dentro del resultado de la jornada de un evento, hago click en el botón volver	El sistema me regresa a la pantalla anterior del evento.
Clickeo uno de los ítems listados	El sistema me muestra la información detallada del partido seleccionado.

<b>Historia de usuario: 30</b>	
<b>Título</b> Crear Partido	
<b>Descripción:</b> yo <b>como</b> usuario Organizador <b>quiero</b> crear un partido para poder llevar a cabo mi evento con el o los partidos que creo	
<b>Pantalla:</b> Figura N.º 146	
<b>Prioridad:</b> 75	<b>Estimación:</b> 30
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Ingreso la fecha y hora de inicio anterior a la fecha actual	El sistema me notifique que debo seleccionar una fecha válida y me sobrescriba la fecha ingresada como la fecha actual
Ingresó un equipo no habilitado	el sistema me notifique que el equipo no está habilitado y no me permita ingresarlo como participante del partido
Ingresó un equipo no vigente	el sistema me notifique que el equipo no está vigente y no me permita ingresarlo como participante del partido

<b>Historia de usuario: 31</b>	
<b>Título:</b> Listar árbitros	
<b>Descripción:</b> Como usuario con rol Organizador <b>quiero</b> visualizar los árbitros disponibles <b>para poder</b> ver qué árbitros disponibles están cargados en el sistema	
<b>Pantalla:</b> Figura N.º 144	
<b>Prioridad:</b> 10	<b>Estimación:</b> 2
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Listo los árbitros	Ver desplegados todos los árbitros disponibles cargados en la plataforma
Listo los árbitros	No debería poder ver la opción de poder editar los árbitros a no ser que tenga el permiso de editar asignado
Listo los árbitros	No debería ver los árbitros con estado inactivo a no ser que seleccione la opción "Ver árbitros con estado inactivo"

<b>Historia de usuario: 32</b>	
<b>Título</b> Alta jornada del evento	
<b>Descripción:</b> Como organizador <b>quiero</b> crear jornadas de partidos <b>para poder</b> setear los equipos, horarios y canchas en las que se juegan.	

<b>Pantalla:</b> Figura N.º 145	
<b>Prioridad:</b> 100	<b>Estimación:</b> 55
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera seleccionar una fecha	Que el sistema verifique que sea después de la fecha inicio evento
Quiera agregar una franja horaria	Que el sistema le permita ingresar franjas horarias Desde-Hasta
Quiera guardar la Jornada	Que el sistema guarde la jornada en conjunto con todas las franjas horarias seleccionadas

Módulo de recomendación

<b>Historia de usuario:</b> 33	
<b>Título:</b> Mostrar eventos recomendados	
<b>Descripción:</b> Como usuario <b>quiero</b> poder visualizar los eventos recomendados <b>para poder</b> inscribirme en algún evento que pueda ser mejor para mi	
<b>Pantalla:</b> Figura N.º 86	
<b>Prioridad:</b> 90	<b>Estimación:</b> 15
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Me encuentre en la pantalla principal	Que el sistema me muestre todos los eventos recomendados a mi perfil según mis deportes favoritos, la ubicación cargada en mi perfil, el promedio del organizador y el tipo de organizador.

Módulo de inscripción

<b>Historia de usuario: 34</b>	
<b>Título:</b> Inscribirse a un Evento como deportista	
<b>Descripción:</b> Como usuario de mi equipo <b>quiero</b> inscribirme a una categoría de un evento <b>para poder</b> dejar constancia de mi participación en el mismo.	
<b>Pantalla:</b> Figura N.º xx	
<b>Prioridad:</b> 100	<b>Estimación:</b> 30
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera inscribirme a una categoría de un evento, pero la fecha actual es mayor a la fecha fin inscripción.	Que el sistema no me permita inscribirme al evento y me notifique que la fecha límite para inscribirse ya pasó.
Quiera inscribirme a una categoría de un evento, pero la fecha actual es menor a la fecha inicio inscripción.	Que el sistema no me permita inscribirme al evento y me notifique la fecha en la que podré inscribirme.
Quiera inscribirme a una categoría de un evento, pero la cantidad de equipos inscriptos es igual a la cantidad máxima permitida en la categoría.	Que el sistema no me permita inscribirme al evento y me notifique que ya no hay espacio en la categoría seleccionada.
Quiera inscribirme a una categoría de un evento, pero la cantidad de deportistas en mi equipo es menor a la cantidad mínima de deportistas permitida por equipo en la categoría.	Que el sistema no me permita inscribirme al evento y me notifique que no hay suficientes deportistas en el equipo.
Quiera inscribirme a una categoría de un evento, pero la cantidad de deportistas en mi equipo es mayor a la cantidad máxima de deportistas permitida por equipo en la categoría.	Que el sistema no me permita inscribirme al evento y me notifique que hay demasiados deportistas en el equipo.
Quiera dar mi preferencia horaria	Que el sistema ofrezca los horarios

	definidos por el organizador a la hora de crear la categoría.
Quiera elegir el método de pago	Que el sistema ofrezca los métodos de pago definidos por el organizador a la hora de crear el evento.
Complete mi inscripción	Que el sistema guarde la inscripción y le asigne el estado "No Pago" mostrándome un cartel con el mensaje "inscripción guardada con éxito"

<b>Historia de usuario: 35</b>	
<b>Título:</b> Darse de baja de un Evento como deportista	
<b>Descripción:</b> Como usuario de mi equipo <b>quiero</b> dar al equipo de baja en una categoría de un evento <b>para poder</b> desvincular a mi equipo de la categoría seleccionada.	
<b>Pantalla:</b> Figura N.º 69	
<b>Prioridad:</b> 80	<b>Estimación:</b> 10
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera darme de baja en una categoría, pero la fecha actual es mayor a la fecha de Fin Inscripción	Que el sistema no me permita darme de baja de la categoría y me notifique que el evento está en una etapa avanzada y no es posible darse de baja.
Quiera darme de baja en una categoría e ingrese una descripción opcional	Que el sistema actualice el estado del equipo a "Inactivo" y que disminuya en 1 la cantidad de equipos que participaron en el evento y me muestre un cartel de éxito.



<b>Historia de usuario: 36</b>	
<b>Título:</b> Aceptar solicitud inscripción a un Evento como Árbitro	
<b>Descripción:</b> Como usuario con rol de Árbitro <b>quiero</b> aceptar la solicitud del deportista para participar en el evento <b>para poder</b> confirmar mi participación en el mismo.	
<b>Pantalla:</b> -	
<b>Prioridad:</b> 30	<b>Estimación:</b> 5
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera aceptar la solicitud de inscripción	Que el sistema asigne la fecha actual a la fecha aceptación de la relación entre el evento y el árbitro

<b>Historia de usuario: 37</b>	
<b>Título:</b> Rechazar solicitud inscripción a un Evento como Árbitro	
<b>Descripción:</b> Como usuario con rol de Árbitro <b>quiero</b> rechazar la solicitud del deportista para participar en el evento <b>para poder</b> evitar la inscripción indeseada de un deportista.	
<b>Pantalla:</b> -	
<b>Prioridad:</b> 30	<b>Estimación:</b> 2
<b>Criterios de aceptación</b>	

Cuando	Espero
Clickee en rechazar la solicitud de inscripción	Que el sistema asigne la fecha actual a la fecha rechazo de la relación entre el evento y el jugador
Clickee en rechazar la solicitud de inscripción y el evento ya esté en curso	Que el sistema no me permita rechazar la solicitud de inscripción y me notifique que el evento ya ha comenzado

<b>Historia de usuario: 38</b>	
<b>Título:</b> Darse de baja en un Evento como Patrocinador	
<b>Descripción:</b> Como usuario con rol Patrocinador <b>quiero</b> eliminar la petición de patrocinio <b>para poder</b> anular mi participación en el evento y dejar constancia de ello.	
<b>Pantalla:</b> -	
<b>Prioridad:</b> 45	<b>Estimación:</b> 2
<b>Criterios de aceptación</b>	
Cuando	Espero
Quiera darme de baja, pero la solicitud de patrocinio ya está aceptada.	Que el sistema no me permita darme de baja del evento y me notifique que el evento ya está en curso.
Quiera darme de baja y la solicitud de patrocinio no está aceptada.	Que el sistema actualice el estado de la petición de patrocinio a "Anulada". Y me muestre un cartel con un mensaje de éxito.

<b>Historia de usuario: 39</b>	
<b>Título:</b> Generar petición patrocinio como Patrocinador	
<b>Descripción:</b> Como usuario con rol Patrocinador <b>quiero</b> generar una petición de patrocinio <b>para poder</b> patrocinar un evento.	
<b>Pantalla:</b> Figura N.º 91	
<b>Prioridad:</b> 65	<b>Estimación:</b> 5
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera patrocinar un evento	Que el sistema me permita seleccionar un evento y poder patrocinarlo
Quiera enviar la petición de patrocinio pero esta este vacía	Que el sistema no me permita enviar la petición
Quiera enviar la petición de patrocinio, pero ya haya una petición enviada y no cancelada al mismo evento	Que el sistema no me permita enviarla y me notifique con un mensaje de: "usted ya ha realizado una petición a este evento"
Quiera enviar una petición de patrocinio a un evento y ya haya enviado antes una al mismo evento y esta ya haya sido rechazada	Que el sistema no me lo permita y me notifique con un mensaje "Usted ya ha enviado una petición y esta ha sido rechazada"
Quiera enviar una petición de patrocinio a un evento y este haya comenzado	Que el sistema no me permita enviar la petición y me notifique que el evento ya ha comenzado
Luego de enviar la solicitud de patrocinio	Que el sistema cree una petición de patrocinio en estado "Creada" y que el organizador la pueda ver en su perfil.

<b>Historia de usuario: 40</b>	
<b>Título:</b> Visualizar Peticiones de patrocinio como patrocinador	
<b>Descripción:</b> Como usuario con rol Patrocinador <b>quiero</b> ver mis peticiones de patrocinio <b>para poder</b> cancelar o modificar una petición.	
<b>Pantalla:</b> Figura N.º 175	
<b>Prioridad:</b> 65	<b>Estimación:</b> 5
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Presione ver peticiones pendientes como patrocinador	Que el sistema me muestre las peticiones de patrocinio pendientes de aceptación más en cada evento junto con un botón de cancelar petición.

<b>Historia de usuario: 41</b>	
<b>Título:</b> Visualizar eventos patrocinados como patrocinador	
<b>Descripción:</b> Como usuario con rol Patrocinador <b>quiero</b> observar mis eventos patrocinados <b>para poder</b> tener registro de los eventos que he patrocinado.	
<b>Pantalla:</b> Figura N.º 173	
<b>Prioridad:</b> 65	<b>Estimación:</b> 5

<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera haga click en mis patrocinios	Que el sistema me muestre los eventos patrocinados junto con un botón de ver.

Módulo de sharing

<b>Historia de usuario: 42</b>	
<b>Título:</b> Compartir evento	
<b>Descripción:</b> Como usuario con rol Deportista, Organizador o Patrocinador <b>quiero</b> compartir un evento <b>para poder</b> difundir el evento en las redes sociales y aumentar su difusión.	
<b>Pantalla:</b> Figura N.º 100	
<b>Prioridad:</b> 25	<b>Estimación:</b> 2
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera compartir un evento	Que el sistema me permita seleccionar entre las redes sociales más conocidas: WhatsApp, Facebook y Twitter
Quiera compartir un evento	Que el sistema me dé la opción de copiar al portapapeles el enlace del evento
Cuando seleccione una opción para compartir el evento	Que el sistema me genere un texto para compartir el evento y me redirija a la red social correspondiente con el texto generado.

<b>Historia de usuario: 43</b>
<b>Título:</b> Compartir Equipo

**Descripción:** Como usuario con rol Deportista, Organizador o Patrocinador **quiero** compartir un Equipo **para poder** difundir el Equipo en las redes sociales y aumentar su popularidad.

**Pantalla:** Figura N.º 179

**Prioridad:** 25

**Estimación:** 2

**Criterios de aceptación**

**Cuando**

**Espero**

Quiera compartir un Equipo

Que el sistema me permita seleccionar entre las redes sociales más conocidas: WhatsApp, Facebook y Twitter.

Quiera compartir un Equipo

Que el sistema me dé la opción de copiar al portapapeles el enlace del Equipo.

Quando seleccione una opción para compartir el Equipo

Que el sistema me genere un texto para compartir el equipo y me redirija a la red social correspondiente con el texto generado.

**Historia de usuario:** 44

**Título:** Vincular red social

**Descripción:** Como usuario **quiero** vincular mis redes sociales a mi perfil **para poder** ser encontrado en mis redes sociales por otros usuarios del sistema.

**Pantalla:** Figuras N.º 178

**Prioridad:** 25

**Estimación:** 2

**Criterios de aceptación**

Cuando	Espero
Clickee en vincular red social en mi perfil	Que el sistema me muestre 4 redes sociales a las que puedo vincularme con un input cada una y sean: Facebook, Twitter, LinkedIn e Instagram.
Quiera vincular nuevas redes sociales	Que el sistema vincule la red social ingresada a mi perfil y la muestre cada vez que visualicen mi perfil
Cambie una red social ya vinculada por un espacio en blanco	Que el sistema desvincule la red social cambiada de mi perfil
Cambie una red social ya vinculada por otra cuenta de la misma red social	Que el sistema sustituya el vínculo de mi cuenta de esa red social
Clickee en el botón cancelar	No se produzcan cambios en las redes sociales vinculadas
Clickee en el botón aceptar	Se guarden los cambios que realice en los campos de las redes sociales a vincular

<b>Historia de usuario:</b> 45	
<b>Título:</b> Compartir perfil usuario	
<b>Descripción:</b> Como usuario con Rol Organizador, Deportista, Patrocinador <b>quiero</b> compartir un perfil <b>para poder</b> difundirlo en las redes sociales	
<b>Pantalla:</b> Figura N.º 100	
<b>Prioridad:</b> 60	<b>Estimación:</b> 30
<b>Criterios de aceptación</b>	
Cuando	Espero
Quiera compartir un Perfil	Que el sistema me permita seleccionar entre las redes sociales más conocidas:



	WhatsApp, Facebook y Twitter
Quiera compartir un Perfil	Que el sistema me dé la opción de copiar al portapapeles el enlace del Perfil
Cuando seleccione una opción para compartir el Perfil	Que el sistema me genere un texto para compartir el Perfil y me redirija a la red social correspondiente con el texto generado

### Módulo de perfil

<b>Historia de usuario:</b> 46	
<b>Título:</b> Dar de alta Cancha como Club.	
<b>Descripción:</b> Como usuario con rol Club <b>quiero</b> dar de alta una cancha para un deporte <b>para poder</b> lograr que los eventos de este deporte puedan realizarse en ese lugar.	
<b>Pantalla:</b> Figura N.º 139	
<b>Prioridad:</b> 100	<b>Estimación:</b> 15
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera dar de alta una cancha y todos los datos estén correctos	El sistema guardé la nueva cancha y el Estado de cancha sea habilitado
Quiera dar de alta una cancha y el dato: capacidad de cancha sea erróneo	El sistema muestra un cartel con el mensaje de error.
Quiera dar de alta una cancha y el dato: nombre de cancha sea erróneo	El sistema muestra un cartel con el mensaje de error.
Quiera dar de alta una cancha y el dato: número de canchas sea erróneo	El sistema muestra un cartel con el mensaje de error.

Quiera dar de alta una cancha y el dato: precio de cancha por día sea erróneo	El sistema muestra un cartel con el mensaje de error.
Quiera dar añadir un deporte a la cancha y el deporte ya haya sido añadido	El sistema no me permite seleccionar dos veces el mismo deporte.
Quiera dar de alta una cancha y un dato obligatorio está vacío	El sistema muestra un cartel con el mensaje de error.

<b>Historia de usuario: 47</b>	
<b>Título:</b> Dar de baja Cancha como Club.	
<b>Descripción:</b> Como usuario con rol Club <b>quiero</b> dar de baja una cancha para un deporte <b>para poder</b> reducir la cantidad de canchas disponibles en mi club	
<b>Pantalla:</b> Figura N° 141	
<b>Prioridad:</b> 70	<b>Estimación:</b> 2
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera dar de baja una cancha y el estado de la cancha es "Inactiva"	Que el sistema me notifique que la cancha ya está dada de baja.
Quiera dar de baja una cancha y el estado de la cancha es "Activo"	Que el sistema cambie el estado de la cancha a "Inactiva"
Quiera dar de baja una cancha y el estado de la cancha es "En Uso"	Que el sistema me notifique que la cancha está en uso por algún evento y me muestre un cartel de confirmación de baja, esta cancha dejara de estar disponible a la hora de crear los partidos

--

<b>Historia de usuario:</b> 48
--------------------------------

<b>Título:</b> Modificar Cancha como Club.
--

<b>Descripción:</b> Como usuario con rol Club <b>quiero</b> modificar las características de una cancha <b>para poder</b> actualizar las características de la misma.
---

<b>Pantalla:</b> Figura N.º 140
---------------------------------

<b>Prioridad:</b> 80	<b>Estimación:</b> 5
----------------------	----------------------

<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera modificar una cancha y todos los datos estén correctos	El sistema guardé la cancha
Quiera modificar una cancha y el dato: capacidad de cancha sea erróneo	El sistema muestra un cartel con el mensaje de error.
Quiera modificar una cancha y el dato: nombre de cancha sea erróneo	El sistema muestra un cartel con el mensaje de error.
Quiera modificar una cancha y el dato: número de canchas sea erróneo	El sistema muestra un cartel con el mensaje de error.
Quiera modificar una cancha y el dato: precio de cancha por día sea erróneo	El sistema muestra un cartel con el mensaje de error.
Quiera dar añadir un deporte a la cancha y el deporte ya haya sido añadido	El sistema muestra un cartel con el mensaje de error.
Quiera modificar una cancha y un dato obligatorio está vacío	El sistema muestra un cartel con el mensaje "El campo es obligatorio" en el campo del dato correspondiente.

--

<b>Historia de usuario:</b> 49	
<b>Título:</b> Listar Canchas como Club	
<b>Descripción:</b> Como usuario con rol Club <b>quiero</b> listar las canchas <b>para poder</b> tener conocimientos de cuáles canchas tengo cargadas en mi club	
<b>Pantalla:</b> Figura N.º 138	
<b>Prioridad:</b> 50	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Listo las canchas	Ver desplegadas todas las canchas cargadas en el club, mostrando el nombre de la cancha, el número de canchas, observaciones y precio de la cancha por hora y también el estado de la cancha
Listo las canchas	Debería poder ver la opción de poder editar las canchas al lado de cada cancha listada
Se listan las canchas	Se renderiza un botón de borrar cancha al lado de cada cancha listada para poder borrarla
Clickeo en el botón de volver	El sistema me regresé a la pantalla anterior

<b>Historia de usuario:</b> 50
<b>Título:</b> Visualizar perfil de deportista
<b>Descripción:</b> Como usuario <b>quiero</b> visualizar el perfil de un deportista <b>para poder</b> ver su nombre, foto de perfil, datos personales y los equipos a los que pertenece.

<b>Pantalla:</b> Figura N.º 99	
<b>Prioridad:</b> 80	<b>Estimación:</b> 20
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
El usuario hace click en algún usuario.	Que el sistema muestre el perfil del usuario, con todos sus datos: nombre, foto de perfil, datos personales y equipos en los que se encuentra.

<b>Historia de usuario:</b> 51	
<b>Título</b> Visualizar perfil del organizador	
<b>Descripción:</b> como usuario quiero poder visualizar el perfil del organizador para poder tomar decisiones respecto a la información visualizada	
<b>Pantalla:</b> Figura N° 99	
<b>Prioridad:</b> 100	<b>Estimación:</b> 20
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Clickeo en el perfil de un organizador desde cualquier pantalla en la que figure	El sistema me muestre la información detallada del Organizador: <ul style="list-style-type: none"> <li>● Nombre</li> <li>● mail</li> <li>● fecha de Alta</li> <li>● Imagen de perfil</li> <li>● teléfono</li> <li>● estado</li> <li>● Tipo</li> <li>● Cantidad de eventos organizados</li> </ul>

--	--

<b>Historia de usuario: 52</b>	
<b>Título</b> Visualizar perfil del patrocinador	
<b>Descripción:</b> como usuario <b>quiero</b> ver el perfil de un patrocinador para poder tomar decisiones con respecto a la información del patrocinador consultado	
<b>Pantalla:</b> Figura N.º 99	
<b>Prioridad:</b> 30	<b>Estimación:</b> 5
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Clickeo en el perfil de un Patrocinador desde cualquier pantalla en la que figure	El sistema me muestre la información detallada del Patrocinador: <ul style="list-style-type: none"> <li>● Nombre</li> <li>● mail</li> <li>● fecha de Alta</li> <li>● Imagen de perfil</li> <li>● teléfono</li> <li>● estado</li> <li>● Tipo</li> <li>● Descripción del negocio</li> <li>● Imagen del negocio</li> </ul> Y un botón de volver.

<b>Historia de usuario: 53</b>	
<b>Título</b> Visualizar perfil del club	

<b>Descripción:</b> como usuario <b>quiero</b> mirar los perfiles del club	
<b>Pantalla:</b> Figura N.º 99	
<b>Prioridad:</b> 90	<b>Estimación:</b> 20
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Clickeo en el perfil de un Club desde cualquier pantalla en la que figure	<p>El sistema me muestre la información detallada del Club:</p> <ul style="list-style-type: none"> <li>● Nombre</li> <li>● mail</li> <li>● fecha de Alta</li> <li>● Imagen de perfil</li> <li>● teléfono</li> <li>● estado</li> <li>● Tipo</li> <li>● Equipos del Club</li> <li>● Localidad</li> <li>● Domicilio</li> <li>● Tabla de Canchas</li> <li>● Cantidad de eventos organizados en el club</li> </ul> <p>y un botón de volver.</p>
Clickeo en el botón de volver	El sistema me regresé a la pantalla anterior

<b>Historia de usuario:</b> 54
<b>Título</b> Visualizar canchas del club
<b>Descripción:</b> como usuario <b>quiero</b> visualizar las canchas de un club <b>para poder</b> saber qué canchas están habilitadas para realizar un evento
<b>Pantalla:</b> Figura N.º 138

<b>Prioridad:</b> 10	<b>Estimación:</b> 7
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Visualizo el perfil de un usuario con el rol Club.	Que el sistema muestre todas las canchas del club.

<b>Historia de usuario: 55</b>	
<b>Título:</b> Listar equipo como deportista o club	
<b>Descripción:</b> Como usuario con rol Deportista o Club <b>quiero</b> visualizar mis equipos <b>para poder</b> gestionarlos de forma rápida y fácil.	
<b>Pantalla:</b> Figura N.º XX	
<b>Prioridad:</b> 40	<b>Estimación:</b> 20
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Presiono para visualizar mis equipos.	Que el sistema me muestra una tabla con información de mis equipos y me permita gestionarlos desde la misma.

<b>Historia de usuario: 56</b>	
<b>Título:</b> Crear equipo como deportista	



**Descripción:** Como usuario con rol Deportista **quiero** crear un equipo **para poder** participar en un evento y dejar constancia de ello.

**Pantalla:** Figura N.º 167

**Prioridad:** 100

**Estimación:** 30

**Criterios de aceptación**

**Cuando**

**Espero**

Indique los integrantes

Que el sistema despliegue un buscador con los deportistas disponibles para invitar al equipo

Ingrese un nombre de equipo ya existente

Que el sistema me notifique que el nombre del equipo ya está tomado.

Ingrese un nombre de un club inexistente

Que el sistema me notifique que el club no existe.

Quiero guardar los datos ingresados.

Que se verifique que los campos: nombre, club, y cantidad de integrantes no estén vacíos.

**Historia de usuario:** 57

**Título** Modificar equipo como deportista

**Descripción:** como usuario con rol Deportista **quiero** modificar el equipo al que pertenezco **para poder** actualizar las características del mismo para futuros eventos

**Pantalla:** Figura N.º 167

**Prioridad:** 55

**Estimación:** 15

<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Seleccione un deportista para cargarlo en el equipo.	Que el sistema añada satisfactoriamente el deportista al equipo con estado "Activo"
Seleccione un deportista para eliminarlo del equipo.	Que el sistema cambie el estado del deportista en el equipo seleccionado a "Inactivo".
Cambie el nombre del equipo a uno ya existente	Que el sistema me notifique que el nombre del equipo ya está tomado.
Quiero guardar los datos ingresados.	Que se verifique que los campos: nombre e integrantes no estén vacíos.

<b>Historia de usuario: 58</b>	
<b>Título</b> Salir del equipo como deportista	
<b>Descripción:</b> como usuario con rol Deportista <b>quiero</b> poder salir de un equipo al cual estoy asignado <b>para poder</b> cambiarme a otro equipo o crear otro equipo	
<b>Pantalla:</b> -	
<b>Prioridad:</b> 35	<b>Estimación:</b> 2
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Cuando elija la opción para salir de un equipo	Que el sistema me muestre un cartel que diga "¿Estás seguro que quieres salir del equipo?" y tenga que seleccionar "Confirmar" para darme de baja en el equipo

Quando confirme luego de seleccionar la opción para salir de un equipo	Que el sistema me elimine del equipo.
Quando elija la opción de salir de un equipo al que no pertenezco	Que el sistema me muestre el mensaje "Ya no estas en este equipo"

<b>Historia de usuario: 59</b>	
<b>Título:</b> Crear equipo como club	
<b>Descripción:</b> Como usuario con rol Club <b>quiero</b> crear un equipo <b>para que el equipo pueda</b> participar en un evento representando al club y dejar constancia de ello.	
<b>Pantalla:</b> Figura N.º 167	
<b>Prioridad:</b> 100	<b>Estimación:</b> 5
<b>Criterios de aceptación</b>	
<b>Quando</b>	<b>Espero</b>
Seleccione los integrantes del equipo	Que el sistema despliegue un buscador con todos los deportistas disponibles en la plataforma y pueda seleccionarlos y agregarlos al equipo
Quiero guardar los datos ingresados.	Que se verifique que los campos: nombre, e integrantes seleccionados no esté vacío

<b>Historia de usuario: 60</b>	
<b>Título</b> Modificar equipo como club	
<b>Descripción:</b> como usuario con rol Club <b>quiero</b> modificar el equipo <b>para poder</b> actualizar o cambiar la información del equipo	

<b>Pantalla:</b> Figura N.º 167	
<b>Prioridad:</b> 70	<b>Estimación:</b> 5
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Seleccione un deportista para cargarlo en el equipo.	Que el deportista se añada al equipo satisfactoriamente
Seleccione un deportista para eliminarlo del equipo.	Que el sistema cambie el estado del deportista en el equipo seleccionado a "Inactivo".
Ingrese un nombre de equipo ya existente	Que el sistema me notifique que el nombre del equipo ya está tomado.
Quiero guardar los datos ingresados.	Que se verifique que los campos: nombre, y cantidad de integrantes no estén vacíos.

<b>Historia de usuario:</b> 61	
<b>Título</b> Eliminar equipo como club	
<b>Descripción:</b> Como usuario con rol Club <b>quiero</b> eliminar un equipo <b>para poder</b> desvincular a los deportistas que pertenecen a ese equipo del club	
<b>Pantalla:</b> -	
<b>Prioridad:</b> 30	<b>Estimación:</b> 2
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiero eliminar un equipo de mi club	Que el sistema ponga la fecha actual en la

	fechaFinVig del equipo y que no se me muestre el equipo a la hora de inscribirse en un torneo.
--	--

<b>Historia de usuario:</b> 62	
<b>Título</b> Modificar perfil como deportista	
<b>Descripción:</b> como usuario con rol Deportista <b>quiero</b> modificar mi perfil <b>para poder</b> cambiar o actualizar los datos de mi perfil	
<b>Pantalla:</b> Figura N.º 158	
<b>Prioridad:</b> 50	<b>Estimación:</b> 5
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiero modificar mi nombre.	Que el sistema verifique que tenga más de 3 caracteres.
Quiero modificar mi apellido.	Que el sistema compruebe que tenga más de 3 caracteres.
Quiero modificar mi DNI.	Que el sistema verifique que tenga entre 7 y 11 caracteres numéricos positivos.
Quiero modificar mi fecha de nacimiento	Que el sistema verifique que sea menor a la fecha actual.
Quiero modificar mi teléfono	Que el sistema solo me permita ingresar números.
Quiera modificar la foto de perfil	Que la foto de perfil sea en un formato de imagen válido
Quiero guardar los datos ingresados.	Que se verifique que los campos: nombre, apellido, no estén vacíos.

Quiero guardar los datos ingresados y todos los datos ingresados son correctos.	Que el sistema guarde los cambios sobre el usuario.
---	---

<b>Historia de usuario: 63</b>	
<b>Título</b> Modificar perfil como club	
<b>Descripción:</b> como usuario con rol Club <b>quiero</b> modificar mi perfil <b>para poder</b> cambiar o actualizar los datos de mi perfil	
<b>Pantalla:</b> Figura N° 166	
<b>Prioridad:</b> 50	<b>Estimación:</b> 5
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera modificar la localidad	Que el sistema despliegue las localidades disponibles
Quiera modificar la provincia	Que el sistema despliegue las provincias disponibles
Quiera modificar el departamento	Que el sistema despliegue los departamentos disponibles
Quiera modificar la razón social	Que el sistema verifique que tenga más de 3 caracteres.
Quiera modificar mi teléfono	Que el sistema solo me permita ingresar números.
Quiera modificar la foto de perfil del club	Que la foto de perfil sea en un formato de imagen válido
Quiera modificar mi dirección	Que el sistema usando API de Google maps encuentre el domicilio en un mapa y extraiga la latitud y longitud del lugar seleccionado

Quiero guardar los datos ingresados.	Que se verifique que los campos: razón Social, teléfono, domicilio, localidad, departamento y provincia, no estén vacíos.
Quiero guardar los datos ingresados.	Que el sistema verifique que los parámetros ingresados estén efectivamente en la base de datos para los datos (provincias, localidades y departamentos).
Quiero guardar los datos ingresados y todos los datos ingresados son correctos.	Que el sistema guarde los cambios sobre el usuario.

<b>Historia de usuario:</b> 64	
<b>Título</b> Modificar perfil como árbitro	
<b>Descripción:</b> como usuario con rol Árbitro <b>quiero</b> modificar mi perfil <b>para poder</b> cambiar o actualizar los datos de mi perfil	
<b>Pantalla:</b> Figura N.º -	
<b>Prioridad:</b> 20	<b>Estimación:</b> 2
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiero modificar mi nombre.	Que el sistema verifique que tenga más de 3 caracteres.
Quiero modificar mi apellido.	Que el sistema verifique que tenga más de 3 caracteres.
Quiero modificar mi teléfono	Que el sistema solo me permita ingresar números.
Quiera modificar la foto de perfil	Que la foto de perfil sea en un formato de

	imagen válido
Quiero guardar los datos ingresados.	Que se verifique que los campos: nombre, apellido, deporte, no estén vacíos.
Quiero guardar los datos ingresados y todos los datos ingresados son correctos.	Que el sistema guarde los cambios sobre el usuario.

<b>Historia de usuario: 65</b>	
<b>Título:</b> Modificar perfil como patrocinador	
<b>Descripción:</b> como usuario con rol de patrocinador <b>quiero</b> modificar mi perfil <b>para poder</b> cambiar o actualizar los datos de mi perfil.	
<b>Pantalla:</b> Figura N.º xx	
<b>Prioridad:</b> 50	<b>Estimación:</b> 2
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiero modificar mi nombre.	Que el sistema verifique que tenga más de 3 caracteres.
Quiero modificar mi CUIT.	Que el sistema verifique que tenga entre 9 y 13 caracteres numéricos positivos.
Quiero modificar mi teléfono	Que el sistema solo me permita ingresar números.
Quiera modificar la foto de perfil	Que la foto de perfil sea en un formato de imagen válido
Quiero guardar los datos ingresados.	Que se verifique que los campos: nombre, CUIT, no estén vacíos.
Quiero guardar los datos ingresados y todos los datos ingresados son correctos.	Que el sistema guarde los cambios sobre el usuario.



<b>Historia de usuario: 66</b>	
<b>Título</b> Modificar perfil como organizador	
<b>Descripción:</b> como usuario con rol Organizador <b>quiero</b> modificar mi perfil <b>para poder</b> cambiar o actualizar los datos de mi perfil.	
<b>Pantalla:</b> Figura N.º 159	
<b>Prioridad:</b> 70	<b>Estimación:</b> 2
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiero modificar mi nombre.	Que el sistema verifique que tenga más de 3 caracteres.
Quiero modificar mi CUIT.	Que el sistema verifique que tenga entre 9 y 13 caracteres numéricos positivos.
Quiero modificar mi teléfono	Que el sistema solo me permita ingresar números.
Quiera modificar la foto de perfil	Que la foto de perfil sea en un formato de imagen válido
Quiero guardar los datos ingresados.	Que se verifique que los campos: nombre, CUIT, no estén vacíos.
Quiero guardar los datos ingresados y todos los datos ingresados son correctos.	Que el sistema guarde los cambios sobre el usuario.

<b>Historia de usuario: 67</b>
<b>Título:</b> Agregar deportes favoritos como deportista

**Descripción:** como usuario con rol Deportista **quiero** seleccionar mis deportes favoritos **para poder** obtener sugerencias personalizadas sobre la base de estos.

**Pantalla:** Figura N.º 157

**Prioridad:** 20

**Estimación:** 3

**Criterios de aceptación**

**Cuando**

Cargo deportes favoritos

**Espero**

Que el sistema guarde estas preferencias y las use para hacer el sistema más personalizado.

**Historia de usuario:** 68

**Título:** Quitar deportes favoritos como deportista

**Descripción:** como usuario con rol Deportista **quiero** eliminar de mis deportes favoritos un deporte **para poder** obtener sugerencias personalizadas en base a estos.

**Pantalla:** Figura N.º 157

**Prioridad:** 20

**Estimación:** 3

**Criterios de aceptación**

**Cuando**

Elimino mis deportes favoritos

**Espero**

Que el sistema guarde estas preferencias y las utilice para hacer el sistema más personalizado.

<b>Historia de usuario:</b> 69	
<b>Título</b> Darse de baja como usuario	
<b>Descripción:</b> como usuario <b>quiero</b> darme de baja <b>para poder</b> inhabilitar mi usuario en el sistema	
<b>Pantalla:</b> -	
<b>Prioridad:</b> 5	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiero darme de baja del sistema	Que el sistema modifique al usuario y le asigne la fecha actual a la fechaFinVigUsuario
Quiero darme de baja del sistema	Que el sistema le asigne al usuario el estado "Inactivo"

<b>Historia de usuario:</b> 70	
<b>Título</b> Quitar rol a otro usuario como administrador	
<b>Descripción:</b> como usuario administrador <b>quiero</b> dar de baja un rol al que pertenece otro usuario para actualizar la información del usuario y brindarle los permisos correctos	
<b>Pantalla:</b> Figura N.º 164	
<b>Prioridad:</b> 5	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	

Cuando	Espero
Estoy en el perfil viendo los roles a los que pertenece el usuario objetivo y quiero dar de baja en un rol, al hacer clic en "quitar rol" en un rol del que forme parte	Que el sistema modifique al usuario y le asigne la fecha actual a la fecha de fin de vigencia a la relación del rol con el usuario y muestre un cartel de "rol quitado con éxito"
Estoy en el perfil viendo los roles a los que pertenece el usuario objetivo y quiero dar de baja en un rol, al hacer clic en "borrar rol" en un rol del que NO forme parte	Que el sistema muestre un cartel de: "El usuario ya no forma parte de ese rol"

<b>Historia de usuario:</b> 71	
<b>Título</b> Quitar rol a mi usuario	
<b>Descripción:</b> como usuario quiero dar de baja un rol al que pertenezco para actualizar mi información de usuario y poseer los permisos correctos	
<b>Pantalla:</b> Figura N.º 163	
<b>Prioridad:</b> 5	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
Cuando	Espero
Estoy en mi perfil viendo los roles a los que pertenezco y quiero darme de baja en un rol, al hacer clic en "quitar rol" en un rol del que forme parte	Que el sistema modifique al usuario y le asigne la fecha actual a la fecha de fin de vigencia a la relación del rol con el usuario y muestre un cartel de "rol quitado con éxito"
Estoy en mi perfil y quiero darme de baja en un rol, al hacer clic en "borrar rol" en un rol del que NO forme parte	Que el sistema muestre un cartel de: "Usted no forma parte de ese rol"

--

<b>Historia de usuario:</b> 72	
<b>Título</b> Añadir rol a mi usuario	
<b>Descripción:</b> como usuario <b>quiero</b> añadir un rol para actualizar mi información de usuario y poseer los permisos correctos	
<b>Pantalla:</b> Figura N.º 162	
<b>Prioridad:</b> 5	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Estoy en mi perfil viendo los roles a los que pertenezco y quiero añadirme a un rol, al hacer clic en “añadir rol” en un rol del que no forme parte	Que el sistema modifique al usuario y le asigne la fecha actual a la fecha de inicio de vigencia a la relación del rol con el usuario y muestre un cartel de “rol añadido con éxito”
Estoy en mi perfil y quiero añadirme en un rol, al hacer clic en “añadir rol” en un rol del que ya forme parte	Que el sistema muestre un cartel de: “Usted ya forma parte de ese rol”

<b>Historia de usuario:</b> 73	
<b>Título</b> Añadir rol a un usuario como administrador	
<b>Descripción:</b> como usuario administrador <b>quiero</b> añadir un rol para actualizar la información de usuario y poseer los permisos correctos	
<b>Pantalla:</b> Figura N.º 165	
<b>Prioridad:</b> 5	<b>Estimación:</b> 1

Criterios de aceptación	
Cuando	Espero
Estoy en mi perfil viendo los roles a los que pertenece y quiero añadir al usuario un rol, al hacer clic en "añadir rol" en un rol del que no forme parte	Que el sistema modifique al usuario y le asigne la fecha actual a la fecha de inicio de vigencia a la relación del rol con el usuario y muestre un cartel de "rol añadido con éxito"
Estoy en mi perfil y quiero añadir al usuario en un rol, al hacer click en "añadir rol" en un rol del que ya forme parte	Que el sistema muestre un cartel de: "el usuario ya forma parte de ese rol"

#### Módulo Parametrización

<b>Historia de usuario: 75</b>	
<b>Título:</b> Alta Deporte	
<b>Descripción:</b> Como usuario con permiso Alta deporte <b>quiero</b> crear un deporte <b>para poder</b> utilizarlo como organizador para crear un evento.	
<b>Pantalla:</b> Figura N.º 107	
<b>Prioridad:</b> 100	<b>Estimación:</b> 5
Criterios de aceptación	
Cuando	Espero
Indique el nombre del deporte	Que el sistema no permita el ingreso de caracteres especiales.
Indique la cantidad de Equipos máximos por cancha	Que el sistema solo me permita ingresar números positivos.
Indique la cantidad de Equipos mínimos por cancha	Que el sistema únicamente me permita ingresar números positivos.

Indique la cantidad de integrantes por equipo	Que el sistema solamente me permita ingresar números positivos.
Quiera guardar el deporte	Que el sistema verifique que el nombre, la cantidad de Equipos máximos/mínimos, la cantidad de integrantes permitidos en un equipo, tipo evento y la descripción del deporte no estén vacíos.
Quiera guardar el deporte y todos los datos son correctos	Que el sistema guarde el deporte.

<b>Historia de usuario: 76</b>	
<b>Título:</b> Baja deporte	
<b>Descripción:</b> Como usuario con permiso Baja Tipo Deporte <b>quiero</b> eliminar un deporte <b>para poder</b> evitar que se realicen eventos con este tipo de deporte.	
<b>Pantalla:</b> Figura N.º 109	
<b>Prioridad:</b> 80	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera eliminar el deporte	Que el sistema modifique al deporte y le asigne la fecha actual a la fechaFinVigTipoDeporte.

<b>Historia de usuario: 77</b>
<b>Título:</b> Modificar Tipo Deporte

**Descripción:** Como usuario con permiso Modificar Deporte **quiero** modificar un deporte **para poder** actualizar su información y que los futuros eventos se realicen sobre las nuevas características del deporte.

**Pantalla:** Figura N.º 108

**Prioridad:** 100

**Estimación:** 3

**Criterios de aceptación**

**Cuando**

**Espero**

Indique el nombre del deporte

Que el sistema no permita el ingreso de caracteres especiales.

Indique la cantidad de Equipos máximos por cancha

Que el sistema solo me permita ingresar números positivos.

Indique la cantidad de Equipos mínimos por cancha

Que el sistema solo me permita ingresar números positivos.

Indique la cantidad de integrantes mínimos permitidos en un equipo

Que el sistema solamente me permita ingresar números positivos.

Indique la cantidad de integrantes máximos permitidos en un equipo

Que el sistema exclusivamente me permita ingresar números positivos.

Quiera guardar el deporte

Que el sistema verifique que el nombre, la cantidad de Equipos máximos/mínimos, la cantidad de integrantes máximos/mínimos permitidos en un equipo, tipo evento y la descripción del deporte no estén vacíos.

Quiera modificar el deporte y todos los datos son correctos

Que el sistema modifique el deporte

**Historia de usuario:** 78

**Título:** Listar deportes



**Descripción:** Como usuario con permiso Listar Deportes **quiero** listar los tipos de deportes **para poder** tener conocimientos de cuáles deportes se han creado en la plataforma.

**Pantalla:** Figura N.º 106

**Prioridad:** 60

**Estimación:** 2

**Criterios de aceptación**

**Cuando**

**Espero**

Listo los deportes

Ver desplegados todos tipos de deportes cargados en la plataforma

Listo los deportes

No debería poder ver la opción de poder editar tipos de deportes a no ser que tenga el permiso de editar asignado

Listo los deportes

No debería ver los tipos de deportes con estado inactivo a no ser que seleccione la opción "Ver tipos de deportes con estado inactivo"

**Historia de usuario:** 79

**Título:** Alta Tipo Organizador

**Descripción:** Como usuario con permiso Alta Tipo Organizador **quiero** crear un tipo de Organizador **para poder** agregar un nuevo plan que los organizadores puedan utilizar.

**Pantalla:** Figura N.º 115

**Prioridad:** 100

**Estimación:** 1

**Criterios de aceptación**

<b>Cuando</b>	<b>Espero</b>
Indique el nombre del Tipo Organizador	Que el sistema no permita el ingreso de caracteres especiales.
Indique el costo del Tipo Organizador	Que el sistema solo me permita ingresar números positivos.
Indique el límite de eventos del Tipo Organizador	Que el sistema solo me permita ingresar números positivos.
Indique el valor Agregado Recomendación del Tipo Organizador	Que el sistema solo me permita ingresar números positivos.
Quiera guardar el Tipo Organizador	Que el sistema verifique que el nombre, el costo y el valor agregado recomendación no estén vacíos.
Quiera guardar el Tipo Organizador y todos los datos son correctos	Que el sistema guarde el tipo Organizador.

<b>Historia de usuario:</b> 80	
<b>Título:</b> Baja Tipo Organizador	
<b>Descripción:</b> Como usuario con permiso Baja Tipo Organizador <b>quiero</b> eliminar un tipo de organizador <b>para poder</b> evitar que se creen nuevos organizadores con el plan seleccionado.	
<b>Pantalla:</b> Figura N.º 117	
<b>Prioridad:</b> 80	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera eliminar el Tipo Organizador	Que el sistema modifique al tipo Organizador y le asigne la fecha actual a la fechaFinVigTipoOrganizador

<b>Historia de usuario: 81</b>	
<b>Título:</b> Modificar Tipo Organizador	
<b>Descripción:</b> Como usuario con permiso Modificar Tipo Organizador <b>quiero</b> modificar un tipo de Organizador <b>para poder</b> modificar un plan existente y permitir que los organizadores se puedan ajustar al mismo.	
<b>Pantalla:</b> Figura N.º 116	
<b>Prioridad:</b> 100	<b>Estimación:</b> 2
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Indique el nombre del Tipo Organizador	Que el sistema no permita el ingreso de caracteres especiales.
Indique el costo del Tipo Organizador	Que el sistema solo me permita ingresar números positivos.
Indique el límite de eventos del Tipo Organizador	Que el sistema solo me permita ingresar números positivos.
Indique el valor Agregado Recomendación del Tipo Organizador	Que el sistema solo me permita ingresar números positivos.
Quiera modificar el Tipo Organizador	Que el sistema verifique que el nombre, el costo y el valor agregado recomendación no estén vacíos.
Quiera modificar el Tipo Organizador y todos los datos son correctos	Que el sistema modifique el tipo Organizador.

<b>Historia de usuario: 82</b>
--------------------------------

<b>Título:</b> Listar Tipo Organizador	
<b>Descripción:</b> Como usuario con permiso Listar Tipo Organizador <b>quiero</b> listar los tipos de organizadores <b>para poder</b> visualizar todos los tipos de organizadores cargados en el sistema	
<b>Pantalla:</b> Figura N.º 114	
<b>Prioridad:</b> 60	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Listo los tipos de organizadores	Ver desplegados todos los tipos de organizadores cargados en la plataforma
Listo los tipos de organizadores	No debería poder ver la opción de poder editar tipos de organizadores a no ser que tenga el permiso de editar asignado
Listo los tipos de organizadores	No debería ver los tipos de deportes con estado inactivo a no ser que seleccione la opción "Ver tipos de organizadores con estado inactivo"

<b>Historia de usuario:</b> 83	
<b>Título:</b> Alta Rol	
<b>Descripción:</b> Como usuario con permiso Alta Rol <b>quiero</b> crear un Rol <b>para poder</b> agregar un nuevo rol y asignarle permisos.	
<b>Pantalla:</b> Figura N.º 119	
<b>Prioridad:</b> 100	<b>Estimación:</b> 3

<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Indique el nombre del Rol	Que el sistema no permita el ingreso de caracteres especiales.
Quiera guardar el Rol y no tenga asignados permisos	Que el sistema no me permita guardar el rol y notifique al usuario que al menos debe tener un permiso asignado.
Quiera guardar el Rol y todos los datos son correctos	Que el sistema guarde el Rol.

<b>Historia de usuario: 84</b>	
<b>Título:</b> Baja Rol	
<b>Descripción:</b> Como usuario con permiso Baja Rol <b>quiero</b> eliminar el Rol <b>para poder</b> evitar que se generen nuevos usuarios con el rol seleccionado.	
<b>Pantalla:</b> Figura N.º 121	
<b>Prioridad:</b> 80	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera eliminar el rol	Que el sistema modifique al Rol y le asigne la fecha actual a la fechaFinVigRol

<b>Historia de usuario: 85</b>
--------------------------------

<b>Título:</b> Modificar Rol	
<b>Descripción:</b> Como usuario con permiso Modificar Rol <b>quiero</b> modificar un Rol <b>para poder</b> actualizar sus características y poder imponerlas a los usuarios con el rol seleccionado.	
<b>Pantalla:</b> Figura N.º 120	
<b>Prioridad:</b> 100	<b>Estimación:</b> 2
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Indique el nombre del Rol	Que el sistema no permita el ingreso de caracteres especiales.
Quiera modificar el Rol y no tenga asignados permisos	Que el sistema no me permita modificar el rol y notifique al usuario que al menos debe tener un permiso asignado.
Quiera guardar el Rol y todos los datos son correctos	Que el sistema guarde el Rol.

<b>Historia de usuario:</b> 86	
<b>Título:</b> Listar Roles	
<b>Descripción:</b> Como usuario con permiso Listar Roles <b>quiero</b> listar los roles <b>para poder</b> visualizar todos los roles cargados en el sistema	
<b>Pantalla:</b> Figura N.º 118	
<b>Prioridad:</b> 60	<b>Estimación:</b> 1

<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Listo los roles	Ver desplegados todos los roles cargados en la plataforma
Listo los roles	No debería poder ver la opción de poder editar roles a no ser que tenga el permiso de editar asignado
Listo los roles	No debería ver los roles con estado inactivo a no ser que seleccione la opción "Ver roles con estado inactivo"

<b>Historia de usuario: 87</b>	
<b>Título:</b> Alta Localidad	
<b>Descripción:</b> Como usuario con permiso Alta Localidad <b>quiero</b> crear una localidad para poder agregar una nueva localidad y que pueda ser utilizada por los clubes.	
<b>Pantalla:</b> Figura N.º 123	
<b>Prioridad:</b> 100	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Indique el nombre de la Localidad	Que el sistema no permita el ingreso de caracteres especiales ni dejar el campo vacío
Quiera guardar la Localidad y no tenga asignado un departamento	Que el sistema no me permita guardar la localidad y notifique al usuario que la localidad debe pertenecer a un departamento.

Quiera guardar la Localidad y todos los datos son correctos	Que el sistema guarde la Localidad.
---	-------------------------------------

<b>Historia de usuario: 88</b>	
<b>Título:</b> Baja Localidad	
<b>Descripción:</b> Como usuario con permiso Baja Localidad <b>quiero</b> eliminar la localidad <b>para poder</b> evitar que se creen nuevos clubes en la localidad seleccionada.	
<b>Pantalla:</b> Figura N.º 125	
<b>Prioridad:</b> 80	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera eliminar la Localidad	Que el sistema modifique la Localidad y le asigne la fecha actual a la fechaFinVigLocalidad

<b>Historia de usuario: 89</b>	
<b>Título:</b> Modificar Localidad	
<b>Descripción:</b> Como usuario con permiso Modificar Localidad <b>quiero</b> modificar una localidad <b>para poder</b> actualizar sus características.	
<b>Pantalla:</b> Figura N.º 124	
<b>Prioridad:</b> 100	<b>Estimación:</b> 1



<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Indique el nombre de la Localidad	Que el sistema no permita el ingreso de caracteres especiales ni dejar el campo vacío.
Quiera modificar la Localidad y no tenga asignado un departamento	Que el sistema no me permita modificar la localidad y notifique al usuario que la localidad debe pertenecer a un departamento.
Quiera modificar la Localidad y todos los datos son correctos	Que el sistema guarde la Localidad.

<b>Historia de usuario: 90</b>	
<b>Título:</b> Listar Localidades	
<b>Descripción:</b> Como usuario con permiso Listar Localidades <b>quiero</b> listar las localidades <b>para poder</b> las localidades que están cargadas en el sistema	
<b>Pantalla:</b> Figura N.º 122	
<b>Prioridad:</b> 60	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Listo las localidades	Ver desplegadas todas las localidades cargadas en la plataforma

<b>Historia de usuario: 91</b>
--------------------------------

<b>Título:</b> Alta Departamento	
<b>Descripción:</b> Como usuario con permiso Alta Departamento <b>quiero</b> crear un departamento <b>para poder</b> agregar un nuevo Departamento y que pueda ser utilizado por las localidades.	
<b>Pantalla:</b> Figura N.º 127	
<b>Prioridad:</b> 100	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Indique el nombre del Departamento	Que el sistema no permita el ingreso de caracteres especiales ni dejar el campo vacío
Quiera guardar el Departamento y no tenga asignado una provincia	Que el sistema no me permita guardar el departamento y notifique al usuario que el departamento debe pertenecer a una provincia.
Quiera guardar el Departamento y todos los datos son correctos	Que el sistema guarde el Departamento.

<b>Historia de usuario:</b> 92	
<b>Título:</b> Baja Departamento	
<b>Descripción:</b> Como usuario con permiso Baja Departamento <b>quiero</b> eliminar el departamento <b>para poder</b> evitar que se generen nuevas localidades con el departamento seleccionado.	
<b>Pantalla:</b> Figura N.º 129	
<b>Prioridad:</b> 80	<b>Estimación:</b> 1

Criterios de aceptación	
Cuando	Espero
Quiera eliminar el Departamento	Que el sistema modifique al Departamento y le asigne la fecha actual a la fechaFinVigDepartamento

<b>Historia de usuario:</b> 93	
<b>Título:</b> Modificar Departamento	
<b>Descripción:</b> Como usuario con permiso Modificar Departamento <b>quiero</b> modificar un departamento <b>para poder</b> actualizar sus características.	
<b>Pantalla:</b> Figura N.º 128	
<b>Prioridad:</b> 100	<b>Estimación:</b> 1
Criterios de aceptación	
Cuando	Espero
Indique el nombre del Departamento	Que el sistema no permita el ingreso de caracteres especiales ni dejar el campo vacío.
Quiera modificar el Departamento y no tenga asignado una provincia	Que el sistema no me permita modificar el departamento y notifique al usuario que el departamento debe pertenecer a una provincia.
Quiera guardar el Departamento y todos los datos son correctos	Que el sistema guarde el Departamento.

--

<b>Historia de usuario:</b> 94	
<b>Título:</b> Listar Departamentos	
<b>Descripción:</b> Como usuario con permiso Listar Departamentos <b>quiero</b> visualizar los Departamentos <b>para poder</b> visualizar qué departamentos están cargados en el sistema	
<b>Pantalla:</b> Figura N.º 126	
<b>Prioridad:</b> 60	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Listo los Departamentos	Ver desplegados todos los Departamentos cargados en la plataforma

<b>Historia de usuario:</b> 95	
<b>Título:</b> Alta Provincia	
<b>Descripción:</b> Como usuario con permiso Alta Provincia <b>quiero</b> crear una provincia <b>para poder</b> agregar una nueva Provincia y que pueda ser utilizada por los departamentos.	
<b>Pantalla:</b> Figura N.º 131	
<b>Prioridad:</b> 100	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>

Indique el nombre de la Provincia	Que el sistema no permita el ingreso de caracteres especiales ni dejar el campo vacío
Quiera guardar la Provincia y no tenga asignado un país	Que el sistema no me permita guardar la provincia y notifique al usuario que la localidad debe pertenecer a un país.
Quiera guardar la Provincia y todos los datos son correctos	Que el sistema guarde la Provincia.

<b>Historia de usuario: 96</b>	
<b>Título:</b> Baja Provincia	
<b>Descripción:</b> Como usuario con permiso Baja Provincia <b>quiero</b> eliminar la provincia <b>para poder</b> evitar que se creen nuevas localidades en la provincia seleccionada.	
<b>Pantalla:</b> Figura N.º 133	
<b>Prioridad:</b> 80	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera eliminar la Provincia	Que el sistema modifique la Provincia y le asigne la fecha actual a la fechaFinVigProvincia

<b>Historia de usuario: 97</b>	
<b>Título:</b> Modificar Provincia	
<b>Descripción:</b> Como usuario con permiso Modificar Provincia <b>quiero</b> modificar una	

Provincia <b>para poder</b> actualizar sus características.	
Pantalla: Figura N.º 132	
Prioridad: 100	Estimación: 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Indique el nombre de la Provincia	Que el sistema no permita el ingreso de caracteres especiales ni dejar el campo vacío
Quiera modificar la Provincia y no tenga asignado un país	Que el sistema no me permita modificar la provincia y notifique al usuario que la localidad debe pertenecer a un país.
Quiera modificar la Provincia y todos los datos son correctos	Que el sistema guarde la Provincia.

Historia de usuario: 98	
Título: Listar Provincias	
Descripción: Como usuario con permiso Listar Provincias <b>quiero</b> visualizar las Provincias <b>para poder</b> ver qué provincias están cargadas en el sistema	
Pantalla: Figura N.º 130	
Prioridad: 60	Estimación: 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Listo las Provincias	Ver desplegadas todas las Provincias

	cargadas en la plataforma
--	---------------------------

<b>Historia de usuario:</b> 99	
<b>Título:</b> Alta País	
<b>Descripción:</b> Como usuario con permiso Alta País <b>quiero</b> crear un país <b>para poder</b> agregar un nuevo País y que pueda ser utilizado por las provincias.	
<b>Pantalla:</b> Figura N°135	
<b>Prioridad:</b> 100	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Indique el nombre del País	Que el sistema no permita el ingreso de caracteres especiales.
Quiera guardar el País y todos los datos son correctos	Que el sistema guarde el País.

<b>Historia de usuario:</b> 100	
<b>Título:</b> Baja País	
<b>Descripción:</b> Como usuario con permiso Baja País <b>quiero</b> eliminar el país <b>para poder</b> evitar que se creen nuevas provincias con el país seleccionado.	
<b>Pantalla:</b> Figura N.º 137	

<b>Prioridad:</b> 80	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiera eliminar el País	Que el sistema modifique al País y le asigne la fecha actual a la fechaFinVigPaís

<b>Historia de usuario:</b> 101	
<b>Título:</b> Modificar País	
<b>Descripción:</b> Como usuario con permiso Modificar País <b>quiero</b> modificar un país <b>para poder</b> actualizar sus características.	
<b>Pantalla:</b> Figura N.º 136	
<b>Prioridad:</b> 100	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Indique el nombre del País	Que el sistema no permita el ingreso de caracteres especiales.
Quiera guardar el País y todos los datos son correctos	Que el sistema guarde el País.

<b>Historia de usuario:</b> 102
---------------------------------



<b>Título:</b> Listar Países	
<b>Descripción:</b> Como usuario con permiso Listar países <b>quiero</b> visualizar los países <b>para poder</b> ver qué países están cargados en el sistema	
<b>Pantalla:</b> Figura N.º 134	
<b>Prioridad:</b> 60	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Listo los países	Ver desplegados todas las Provincias cargadas en la plataforma

<b>Historia de usuario:</b> 103	
<b>Título:</b> Alta categoría	
<b>Descripción:</b> Como usuario con rol Organizador <b>quiero</b> crear una categoría <b>para poder</b> utilizarla en los eventos	
<b>Pantalla:</b> Figura N.º 156	
<b>Prioridad:</b> 100	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Indique el nombre de la categoría	Que el sistema no permita el ingreso de caracteres especiales.
Indique la edad mínima	Que el sistema comprueba que la misma

	sea un número positivo.
Indique la edad máxima	Que el sistema verifique que la misma sea un número positivo mayor que la edad mínima.
Indique la cantidad de integrantes mínima	Que el sistema revise que la misma sea un número positivo.
Indique la cantidad de integrantes máxima	Que el sistema compruebe que la misma sea un número positivo mayor que la cantidad de integrantes mínima.
Quiera guardar una categoría sin deporte	Que el sistema no guarde la categoría y me informe del error.
Quiera guardar la categoría y todos los datos son correctos	Que el sistema guarde la categoría.

<b>Historia de usuario:</b> 104	
<b>Título:</b> Modificar Categoría	
<b>Descripción:</b> Como usuario con rol Organizador <b>Quiero</b> modificar una Categoría <b>para poder</b> actualizar sus características.	
<b>Pantalla:</b> Figura N.º 156	
<b>Prioridad:</b> 100	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Indique el nombre de la categoría	Que el sistema no permita el ingreso de caracteres especiales.
Indique la edad mínima	Que el sistema compruebe que la misma sea un número positivo.
Indique la edad máxima	Que el sistema verifique que la misma sea

	un número positivo mayor que la edad mínima.
Indique la cantidad de integrantes mínima	Que el sistema compruebe que la misma sea un número positivo.
Indique la cantidad de integrantes máxima	Que el sistema verifique que la misma sea un número positivo mayor que la cantidad de integrantes mínima.
Quiera guardar una categoría sin deporte	Que el sistema no guarde la categoría y me informe del error.
Quiera guardar la categoría y todos los datos son correctos	Que el sistema guarde la categoría.

<b>Historia de usuario:</b> 105	
<b>Título:</b> Listar categorías	
<b>Descripción:</b> Como usuario con rol administrador <b>quiero</b> visualizar las categorías <b>para poder</b> ver qué categorías puedo usar en mis eventos.	
<b>Pantalla:</b> Figura N.º 155	
<b>Prioridad:</b> 60	<b>Estimación:</b> 1
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Listo las categorías	Que se me muestran todas las categorías que he creado junto con sus atributos más relevantes.

Módulo de reporte

**Historia de usuario:** 106

**Título:** Ver mejores equipos del evento

**Descripción:** Como usuario **quiero** que al visualizar un evento en estado finalizado pueda ver los equipos que obtuvieron mejores puntajes **para** conocer los mejores equipos del evento

**Pantalla:** Figura N.º 174

**Prioridad:** 10

**Estimación:** 8

**Criterios de aceptación**

**Cuando**

**Espero**

Visualizo los detalles de un evento

Poder ver el mejor equipo del evento

Visualizo los detalles de un evento

Que el evento tenga al menos un equipo ganador

**Historia de usuario:** 107

**Título:** Ver deportistas inscriptos

**Descripción:** Como usuario con rol organizador **quiero** poder ver los deportistas suscritos a mi evento **para** tener la información de que personas están suscritas al evento

**Pantalla:** Figura N.º 86

**Prioridad:** 30

**Estimación:** 3

**Criterios de aceptación**

<b>Cuando</b>	<b>Espero</b>
Visualizar las suscripciones a mi evento	Ver todos los equipos que se han inscripto al torneo.

<b>Historia de usuario:</b> 108	
<b>Título:</b> Ver tipos de organizadores	
<b>Descripción:</b> Como usuario con rol administrador <b>quiero</b> poder ver los tipos de organizadores del sistema <b>para</b> obtener información útil respecto de las finanzas del sistema.	
<b>Pantalla:</b> Figura N.º 114	
<b>Prioridad:</b> 30	<b>Estimación:</b> 3
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Visualizo los tipos de organizadores del sistema	Poder visualizar en un gráfico de barras la cantidad de cada uno de los tipos de organizadores, en el año actual.

Módulo de feedback

<b>Historia de usuario:</b> 109	
<b>Título:</b> Realizar un feedback	
<b>Descripción:</b> Como deportista <b>quiero</b> realizar un feedback <b>para poder</b> calificar un evento en el que participe.	
<b>Pantalla:</b> Figura N.º 154 y 153	
<b>Prioridad:</b> 100	<b>Estimación:</b> 2
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Se encuentra en la página principal del evento en que participó	Que el sistema muestra toda la información del evento.
El deportista quiera realizar un feedback	Que el sistema verifique la participación del mismo al evento sea válida. Se muestra en la pantalla la posibilidad de clasificar como estuvo el evento y el club.

Módulo de Pago

<b>Historia de usuario:</b> 110	
<b>Título:</b> Realizar pago por Mercado Pago	
<b>Descripción:</b> Como deportista <b>quiero</b> realizar un pago por mercado pago de inscripción a evento del equipo <b>para poder</b> inscribirnos y participar con mi equipo en el evento	
<b>Pantalla:</b> Figura N.º 151	
<b>Prioridad:</b> 70	<b>Estimación:</b> 35
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Me encuentro en la pantalla de inscripción del evento, selecciono el equipo que quiero inscribir y selecciono pagar con mercado pago.	El sistema me muestre un modal con la información de lo que estoy pagando y me deje seleccionar mi cuenta de mercado pago para pagar.

<b>Historia de usuario:</b> 111	
<b>Título:</b> Realizar pago en Efectivo	
<b>Descripción:</b> Como usuario con rol deportista <b>quiero</b> realizar un pago en efectivo de inscripción a evento del equipo <b>para poder</b> inscribirnos y participar con mi equipo en el evento	
<b>Pantalla:</b> Figura N.º 89	
<b>Prioridad:</b> 95	<b>Estimación:</b> 5

<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Me encuentro en la pantalla de inscripción del evento, selecciono el equipo que quiero inscribir y selecciono pagar en efectivo.	El sistema me inscriba en el evento con el estado no pagado.

<b>Historia de usuario: 112</b>	
<b>Título:</b> Confirmar pago como organizador o árbitro	
<b>Descripción:</b> Como Organizador / Árbitro <b>quiero</b> Confirmar un pago realizado en efectivo de una inscripción de un equipo <b>para poder</b> Inscribir el equipo en mi evento.	
<b>Pantalla:</b> Figura N.º 152	
<b>Prioridad:</b> 70	<b>Estimación:</b> 3
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Clickeo en el botón de confirmar pago dentro de la lista de equipos inscriptos dentro de mi evento	Que el sistema guarde el estado de la inscripción del equipo como "pagado" y me muestre un cartel con el mensaje "Pago confirmado" y me devuelva a la lista de equipos inscriptos

<b>Historia de usuario: 113</b>	
<b>Título:</b> Cancelar pago como organizador o árbitro	



**Descripción:** Como Organizador / Árbitro **quiero** cancelar un pago realizado en efectivo de una inscripción de un equipo **para poder** dejar constancia que el equipo no pagó inscripción.

**Pantalla:** Figura N.º 152

**Prioridad:** 65

**Estimación:** 2

**Criterios de aceptación**

**Cuando**

Clicqueo en el botón de Rechazar pago a un pago ya confirmado dentro de la lista de equipos inscriptos dentro de mi evento

**Espero**

Que el sistema guarde el estado de la inscripción del equipo como "Reservado para pago" y me muestre un cartel con el mensaje "Pago a Confirmar" y me devuelva a la lista de equipos inscriptos

**Historia de usuario:** 114

**Título:** Expulsar Equipo como Organizador

**Descripción:** Como usuario con rol Organizador **quiero** Expulsar equipo **para poder** continuar adecuadamente con el evento.

**Pantalla:** -

**Prioridad:** 25

**Estimación:** 3

**Criterios de aceptación**

**Cuando**

Clickeo en el botón de Expulsar Equipo dentro de la lista de equipos inscriptos dentro de mi evento

**Espero**

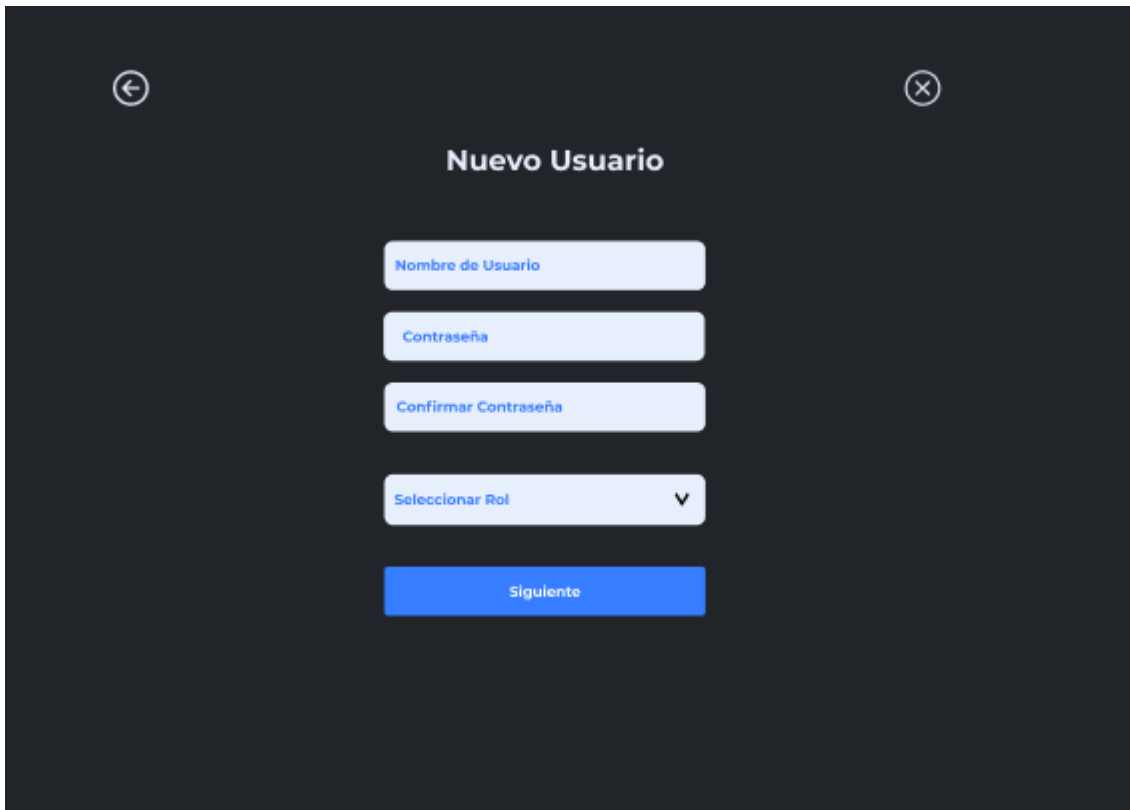
Que el sistema desvincule el equipo de mi evento.

Clickeo en el botón de Expulsar Equipo dentro de la lista de equipos inscriptos dentro de mi evento y el evento ya ha finalizado	Que el sistema me muestre un cartel de error diciendo que el evento ya ha finalizado
Clickeo en el botón de Expulsar Equipo dentro de la lista de equipos inscriptos dentro de mi evento y el equipo ya no esté inscripto	Que el sistema me muestre un mensaje de error diciendo que el equipo ya no está inscripto.

### Módulo de publicidad

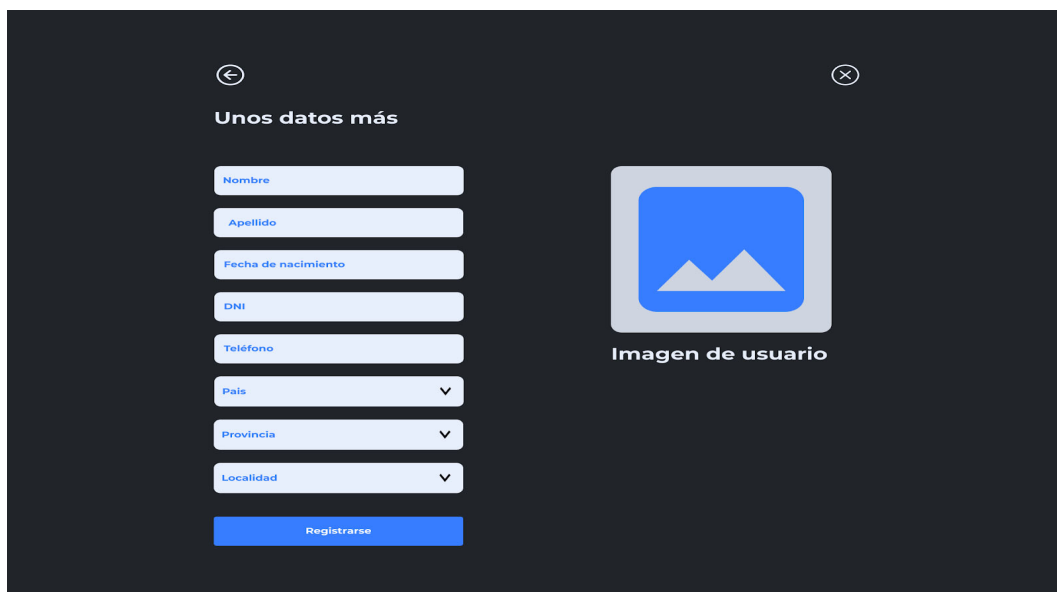
<b>Historia de usuario:</b> 115	
<b>Título:</b> Establecer publicidad	
<b>Descripción:</b> Yo <b>como</b> administrador <b>quiero</b> establecer la publicidad que se muestra en el sistema <b>para poder</b> gestionar de forma sencilla la publicidad que entra en la aplicación.	
<b>Pantalla:</b> Figura N.º 105	
<b>Prioridad:</b> 65	<b>Estimación:</b> 30
<b>Criterios de aceptación</b>	
<b>Cuando</b>	<b>Espero</b>
Quiero cargar una publicidad de una marca en específico.	Que el sistema me deje cargar un banner de publicidad con un link hacia donde se redirigirá a los usuarios en el caso que presionen sobre la misma. Además, el sistema me debería dejar el lugar donde se posicionará la publicidad.

## 4. Pantallas



The screenshot shows a registration form titled "Nuevo Usuario" on a dark background. At the top left is a back arrow icon and at the top right is a close 'X' icon. The form contains the following elements from top to bottom: a text input field labeled "Nombre de Usuario", a text input field labeled "Contraseña", a text input field labeled "Confirmar Contraseña", a dropdown menu labeled "Seleccionar Rol" with a downward arrow, and a blue button labeled "Siguiete".

Figura 65: Pantalla inicial de registro



The screenshot shows a registration form titled "Unos datos más" on a dark background. At the top left is a back arrow icon and at the top right is a close 'X' icon. The form contains the following elements from top to bottom: a text input field labeled "Nombre", a text input field labeled "Apellido", a text input field labeled "Fecha de nacimiento", a text input field labeled "DNI", a text input field labeled "Teléfono", a dropdown menu labeled "País", a dropdown menu labeled "Provincia", a dropdown menu labeled "Localidad", and a blue button labeled "Registrarse". To the right of the form is a square placeholder for a user profile picture, labeled "Imagen de usuario", which contains a blue icon of a mountain range.

Figura 66: Pantalla de registro de deportista

←

⊗

### Unos datos más

Razon social

CUIT

Domicilio social

Teléfono

Registrarse

Imagen de usuario

Figura 67: Pantalla de registro de patrocinador

←

✕

### Unos datos más

Razon Social

Domicilio

Fecha de nacimiento

DNI

Teléfono

País ▼

Provincia ▼

Localidad ▼

Registrarse

Imagen de Club

Ubicación del Club

Figura 68: Pantalla de registro de club

Unos datos más

Nombre

Apellido

Fecha de nacimiento

DNI

Teléfono

País ▼

Provincia ▼

Localidad ▼

Facebook

Instagram

linkedin

Twitter

Registrarse

Imagen de usuario

Figura 69: Pantalla de registro de organizador

Unos datos más

Nombre

Apellido

Fecha de nacimiento

DNI

Teléfono

País

Provincia

Localidad

Facebook

Instagram

linkedin

Twitter

Imagen de usuario

Registrarse

Figura 70: Pantalla de registro de árbitro

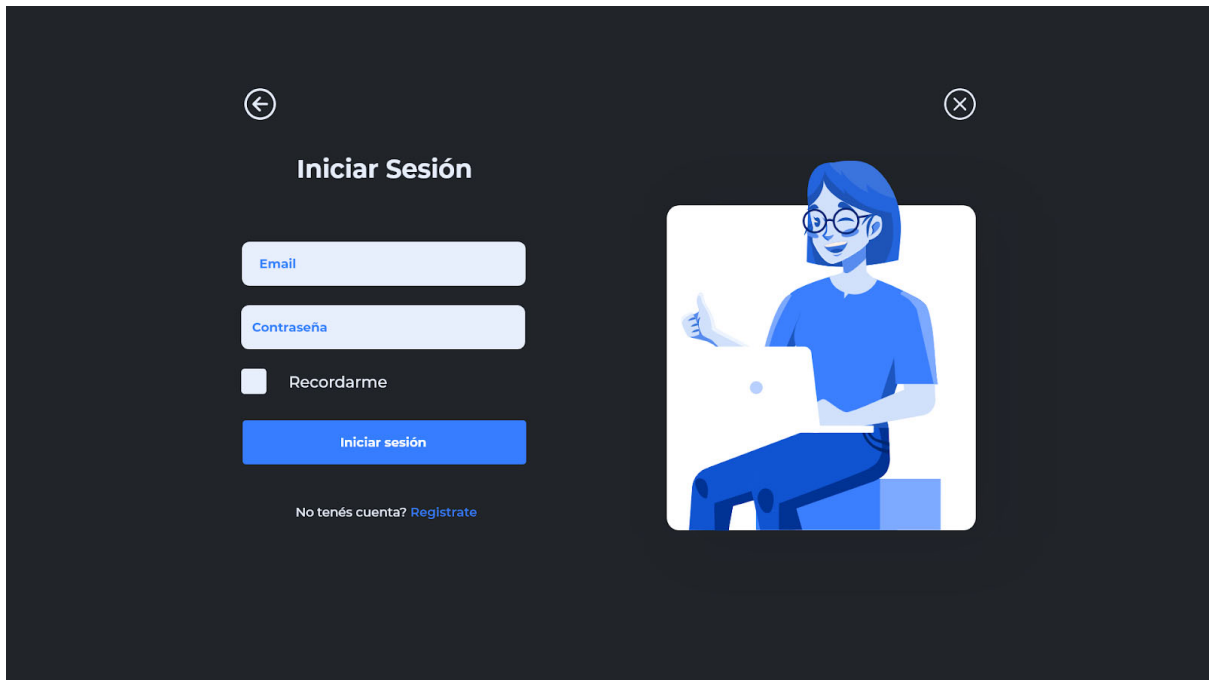


Figura 71: Pantalla de inicio de sesión

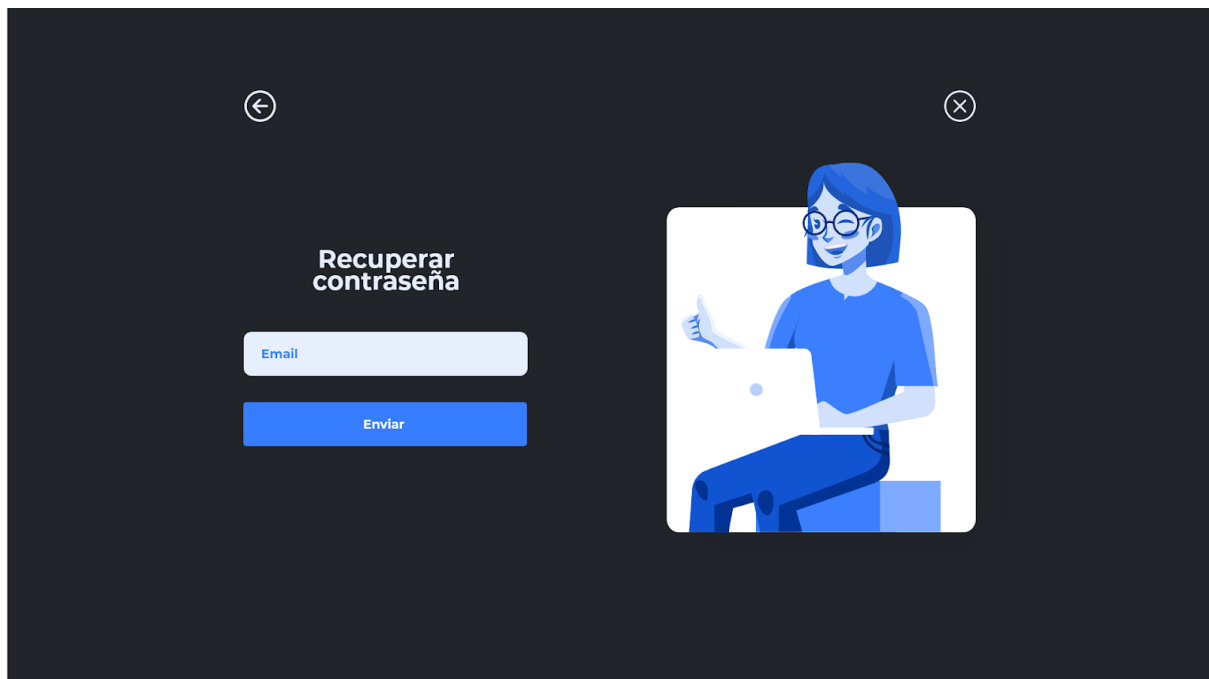
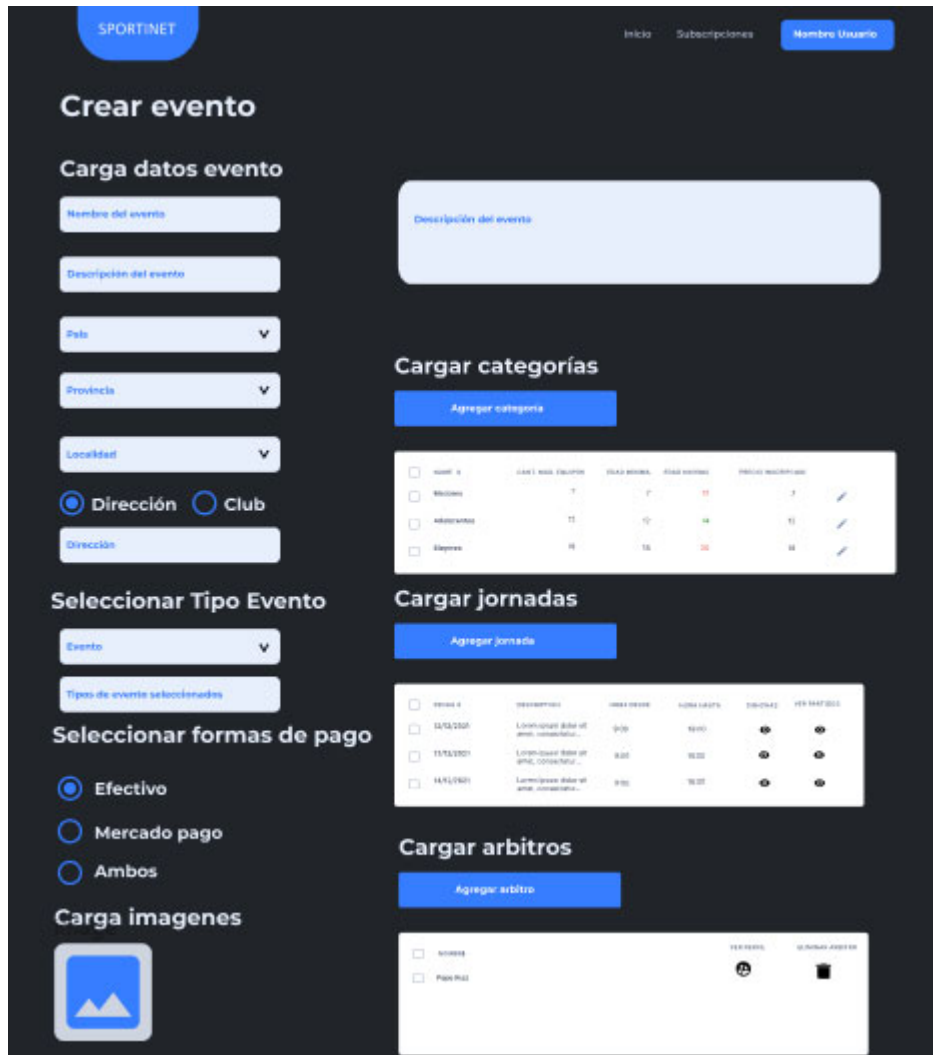


Figura 72: Pantalla de recuperación de contraseña





**SPORTINET** Inicio Subscripciones Nombre Usuario

### Crear evento

**Carga datos evento**

Nombre del evento

Descripción del evento

País

Provincia

Localidad

Dirección  Club

Dirección

**Seleccionar Tipo Evento**

Evento

Tipo de evento seleccionados

**Seleccionar formas de pago**

Efectivo

Mercado pago

Ambos

**Carga imagenes**

**Cargar categorías**

Agregar categoría

id	DESCRIPCIÓN	FECHA INICIAL	FECHA FINAL	FECHA INICIAL	FECHA FINAL
1	...	...	...	...	...
2	...	...	...	...	...
3	...	...	...	...	...

**Cargar jornadas**

Agregar jornada

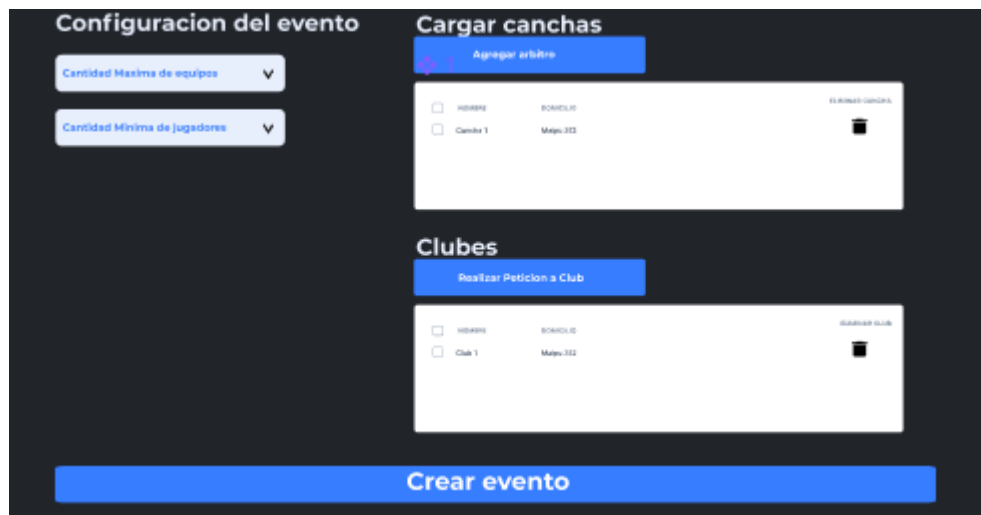
FECHA	DESCRIPCIÓN	HORA INICIAL	HORA FINAL	DELEGADO	VER DETALLES
11/01/2021	...	...	...	...	...
12/01/2021	...	...	...	...	...
13/01/2021	...	...	...	...	...

**Cargar arbitros**

Agregar arbitro

FECHA	DESCRIPCIÓN	HORA INICIAL	HORA FINAL	DELEGADO	VER DETALLES
...	...	...	...	...	...

Figura 73: Pantalla 1 de creación de evento



**Configuración del evento**

Cantidad Máxima de equipos

Cantidad Mínima de jugadores

**Cargar canchas**

Agregar cancha

FECHA	DESCRIPCIÓN	HORA INICIAL	HORA FINAL	DELEGADO	VER DETALLES
...	...	...	...	...	...

**Clubes**

Realizar Petición a Club

FECHA	DESCRIPCIÓN	HORA INICIAL	HORA FINAL	DELEGADO	VER DETALLES
...	...	...	...	...	...

**Crear evento**

Figura 74: Pantalla 2 de creación de evento



**Modificar Evento**

**Carga datos evento**

Nombre del evento:

Descripción del evento:

Fecha:

Horario:

Lugar:

Dirección  Club

Estado:

**Seleccionar Tipo Evento**

Evento:

Tipo de evento seleccionado:

**Seleccionar formas de pago**

Efectivo

Mercado pago

Ambos

**Carga imágenes**

**Configuración del evento**

Cantidad máxima de equipos:

Cantidad máxima de jugadores:

**Cargar categorías**

Añadir categoría

ID	Nombre	Color	Activo
1	...	...	...
2	...	...	...
3	...	...	...

**Cargar jornadas**

Añadir jornada

ID	Nombre	Fecha	Horario	Estado
1	...	...	...	...
2	...	...	...	...
3	...	...	...	...

**Cargar árbitros**

Añadir árbitro

ID	Nombre	Activo
1	...	...
2	...	...

**Cargar canchas**

Añadir cancha

ID	Nombre	Activo
1	...	...
2	...	...

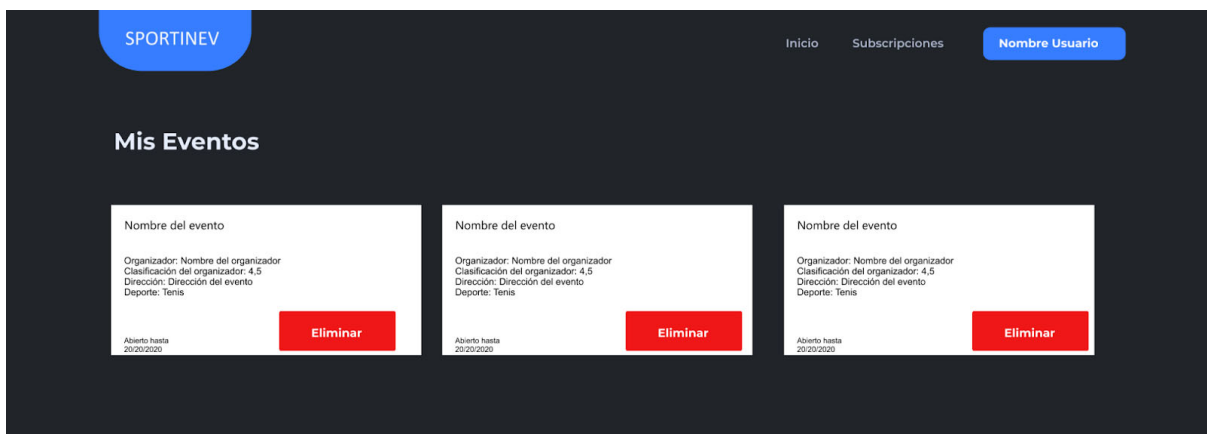
**Clubes**

Añadir club

ID	Nombre	Activo
1	...	...
2	...	...

**Guardar evento**

Figura 75: Pantalla de modificación de evento



**SPORTINEV** Inicio Subscripciones Nombre Usuario

**Mis Eventos**

Nombre del evento

Organizador: Nombre del organizador  
 Clasificación del organizador: 4,5  
 Dirección: Dirección del evento  
 Deporte: Tenis

Abierto hasta: 20/20/2020 **Eliminar**

Nombre del evento

Organizador: Nombre del organizador  
 Clasificación del organizador: 4,6  
 Dirección: Dirección del evento  
 Deporte: Tenis

Abierto hasta: 20/20/2020 **Eliminar**

Nombre del evento

Organizador: Nombre del organizador  
 Clasificación del organizador: 4,5  
 Dirección: Dirección del evento  
 Deporte: Tenis

Abierto hasta: 20/20/2020 **Eliminar**

Figura 76: Pantalla de eliminación de evento

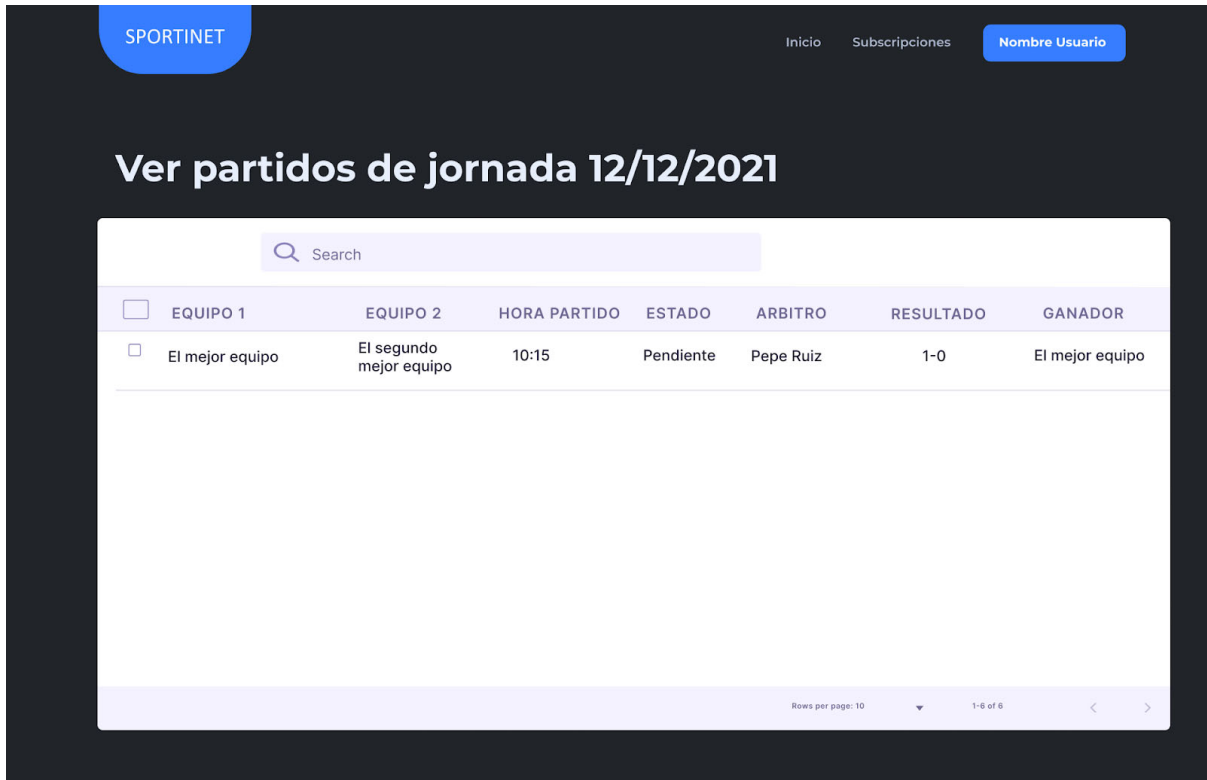


Figura 77: Pantalla de visualización de resultados de partidos de jornada

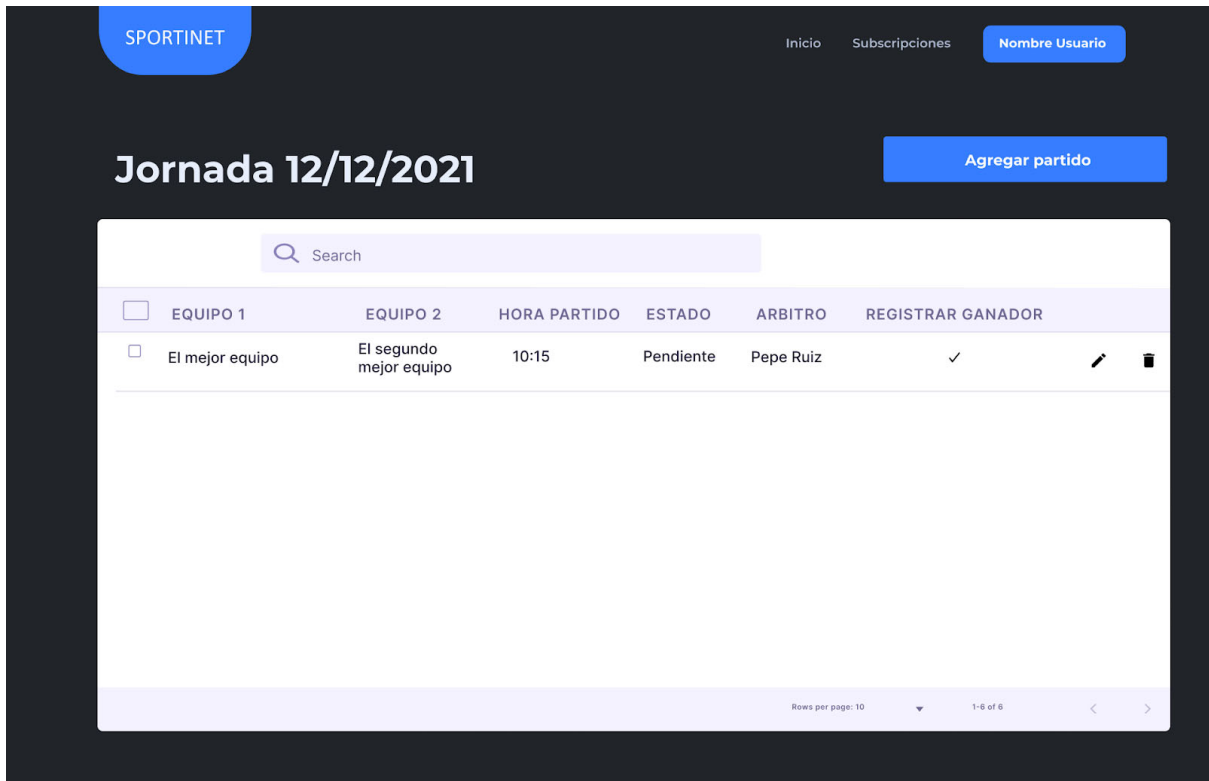


Figura 78: Pantalla de visualización de jornada como organizador

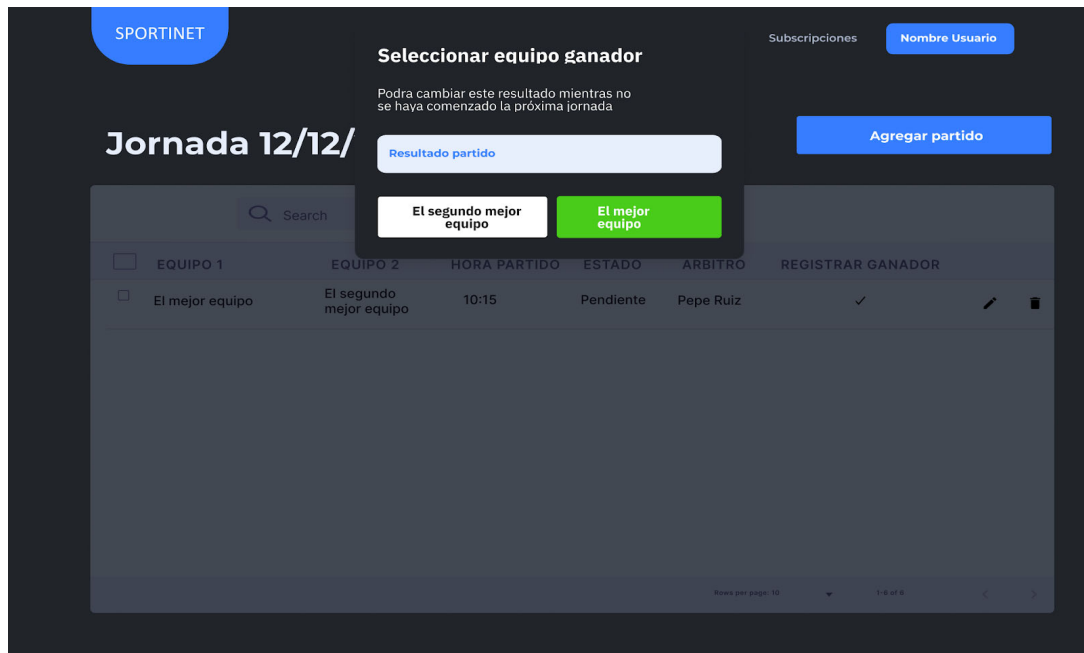


Figura 79: Pantalla de registro de resultado de partido

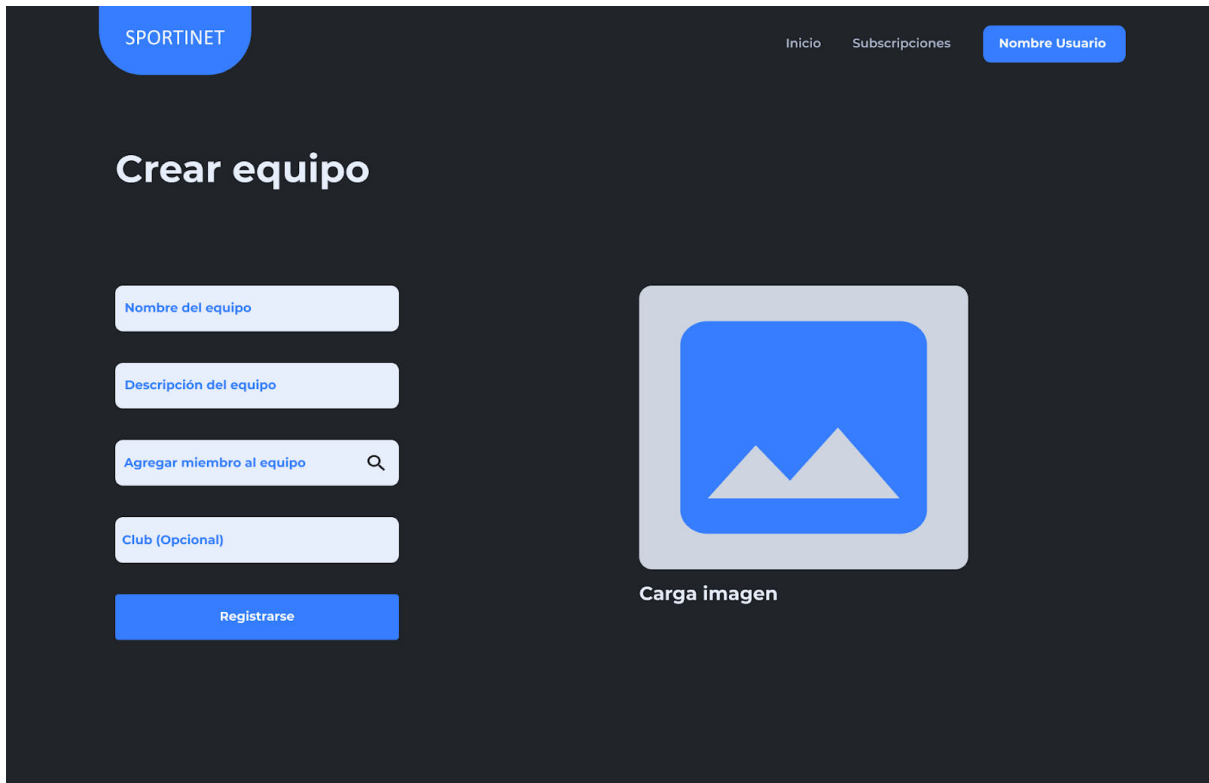


Figura 80: Pantalla de creación de equipo como deportista

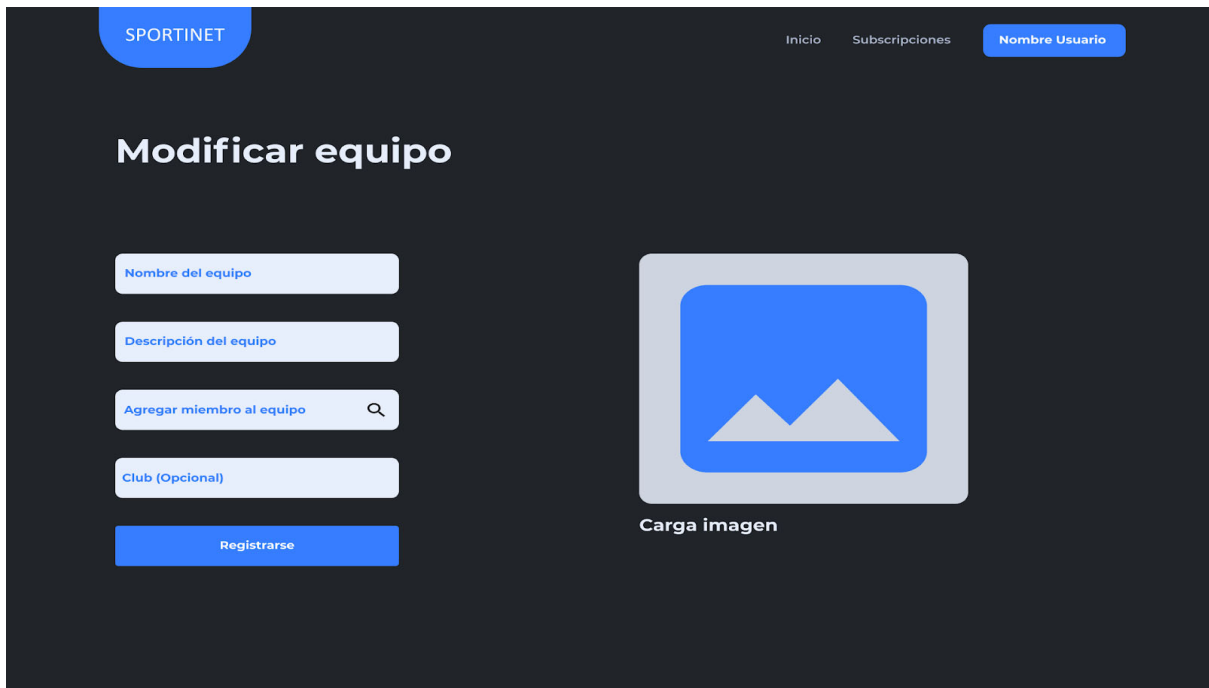


Figura 81: Pantalla de modificación de equipo

The screenshot displays the 'Solicitudes de eventos' (Event Requests) page. At the top, there is a search bar with the placeholder text 'Search'. Below the search bar is a table with the following columns: 'NOMBRE ORGANIZADOR', 'VER PERFIL ORGANIZADOR', 'NOMBRE EVENTO', 'CATEGORIA EVENTO', and 'VER DETALLE EVENTO'. The table contains two rows of data, each with a checkbox on the left and an eye icon on the right. The first row shows 'Lorem ipsum' as the organizer, 'El mejor evento del mundo' as the event name, and 'Primera' as the category. The second row shows 'Lorem ipsum' as the organizer, 'El mejor evento del mundo' as the event name, and 'Segunda' as the category. At the bottom of the table, there is a pagination control showing 'Rows per page: 10' and '1-6 of 6'.

Figura 82: Pantalla de visualización de solicitudes de eventos como club

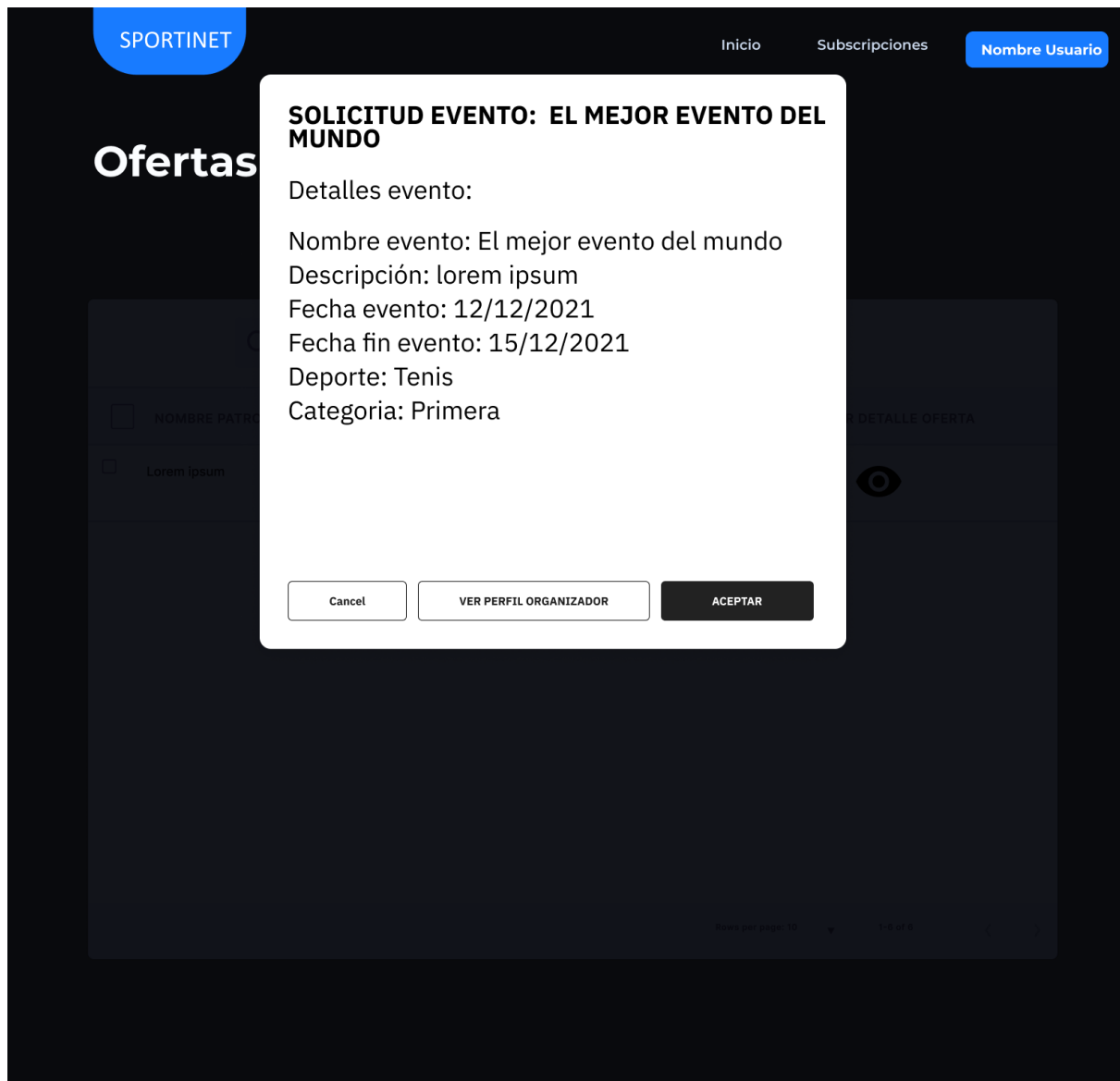


Figura 83: Pantalla para aceptar/rechazar petición evento como Club

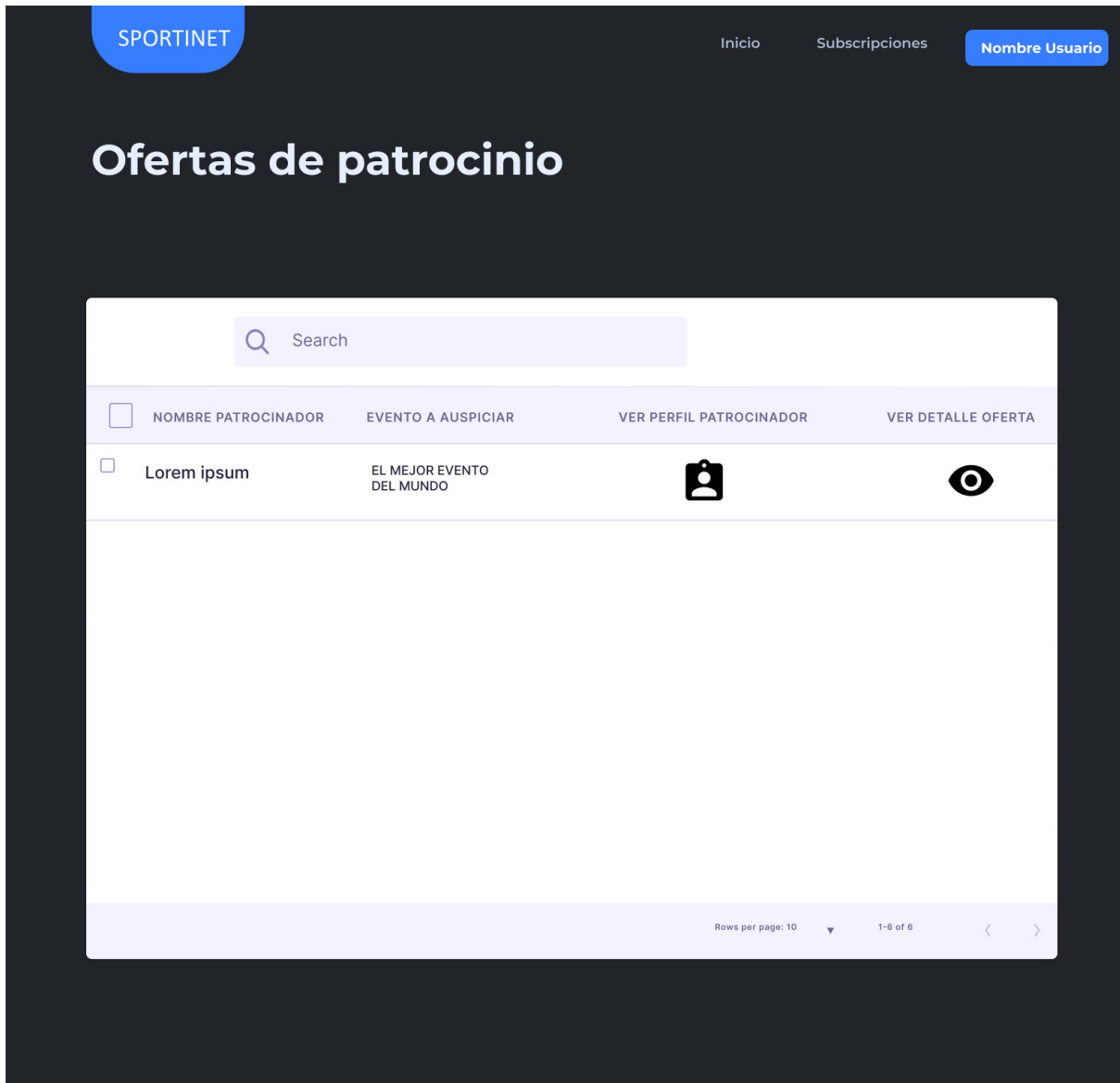


Figura 84: Pantalla de visualización de ofertas de patrocinio como Organizador



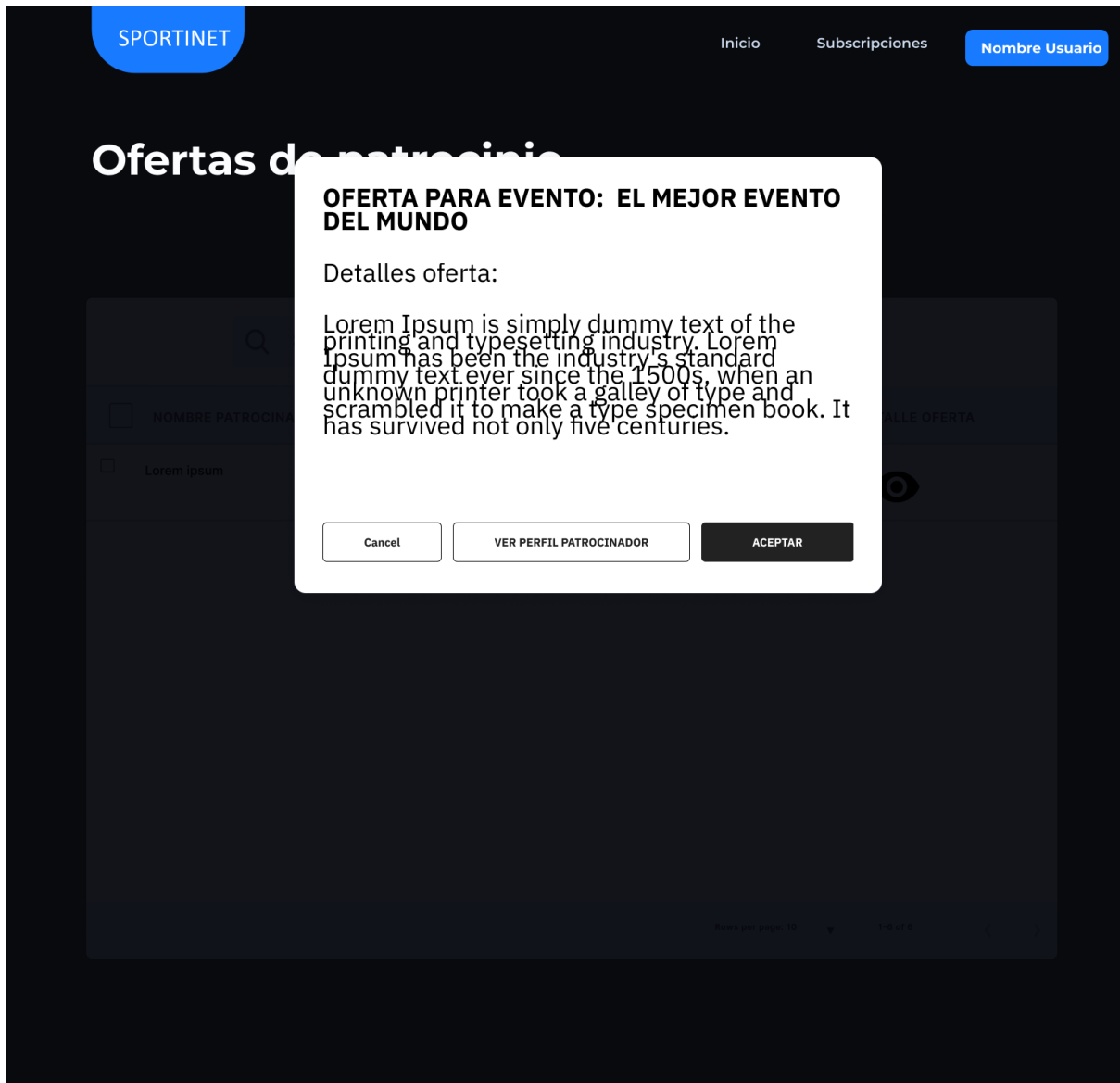


Figura 85: Pantalla para aceptar/rechazar oferta de patrocinio como Organizador

The screenshot shows a dark-themed user interface for an event. At the top, there is a navigation bar with 'SPORTINET' on the left and 'Inicio', 'Subscripciones', and 'Nombre Usuario' on the right. The main heading is 'EI MEJOR EVENTO DEL MUNDO'. Below it is a blue square icon with a white mountain range. Two blue buttons, 'COMPARTIR' and 'INSCRIBIRSE', are positioned below the icon. The 'Información' section follows, with a sub-heading 'Descripción del evento' and a paragraph of Lorem Ipsum text. Below this, it states 'Cantidad de personas inscriptas en el evento: 100'. The 'Ubicación del evento' section features a map of Birmingham, Alabama. The 'Patrocinadores del evento' section displays the logos for Nike and Adidas. Finally, the 'Organizador del evento' section shows a profile for 'Luisa Perez' with a circular avatar and a five-star rating.

Figura 86: Pantalla de visualización de evento como deportista

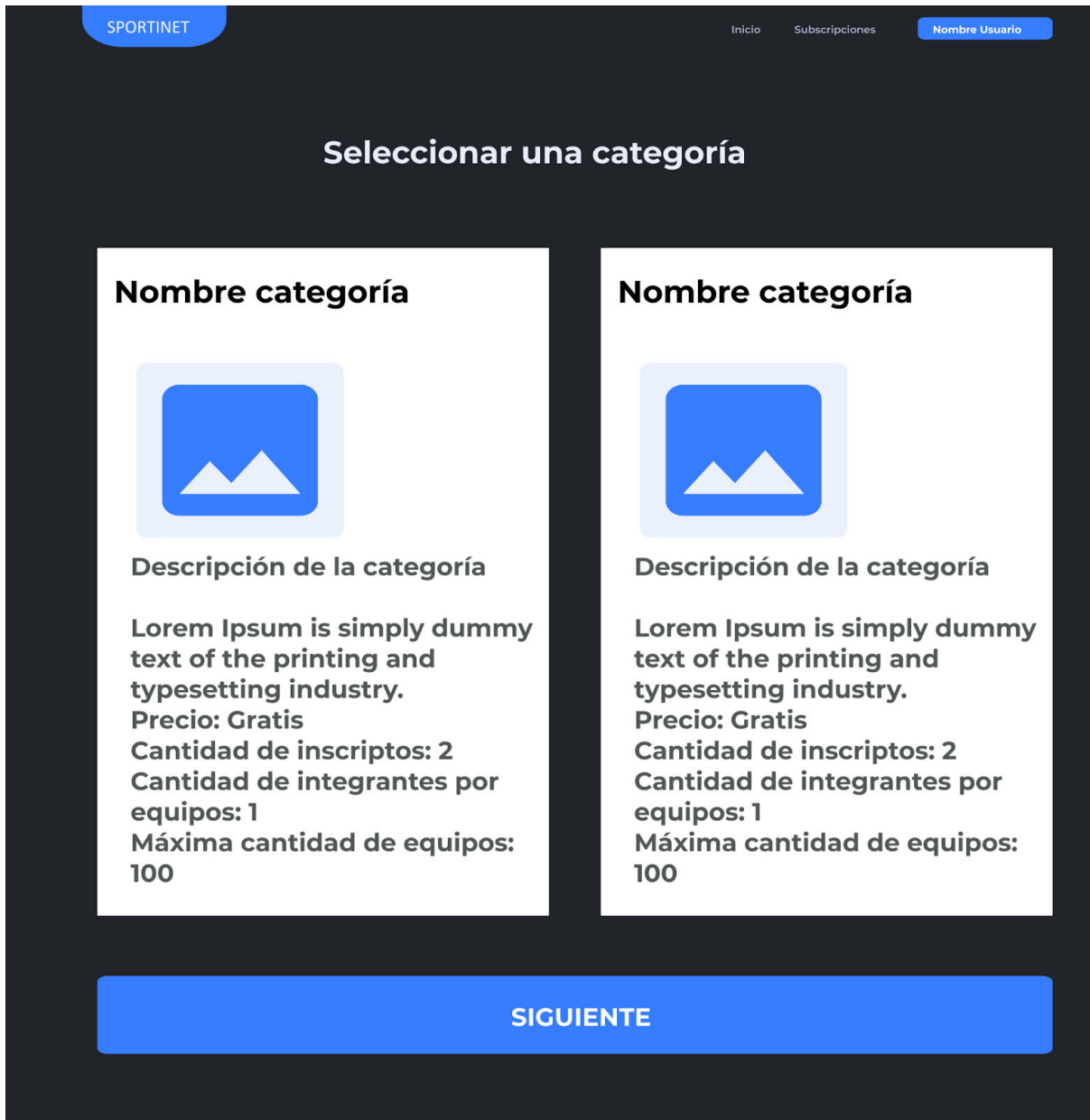


Figura 87: Pantalla de selección de categoría a la que se quiere inscribir en evento

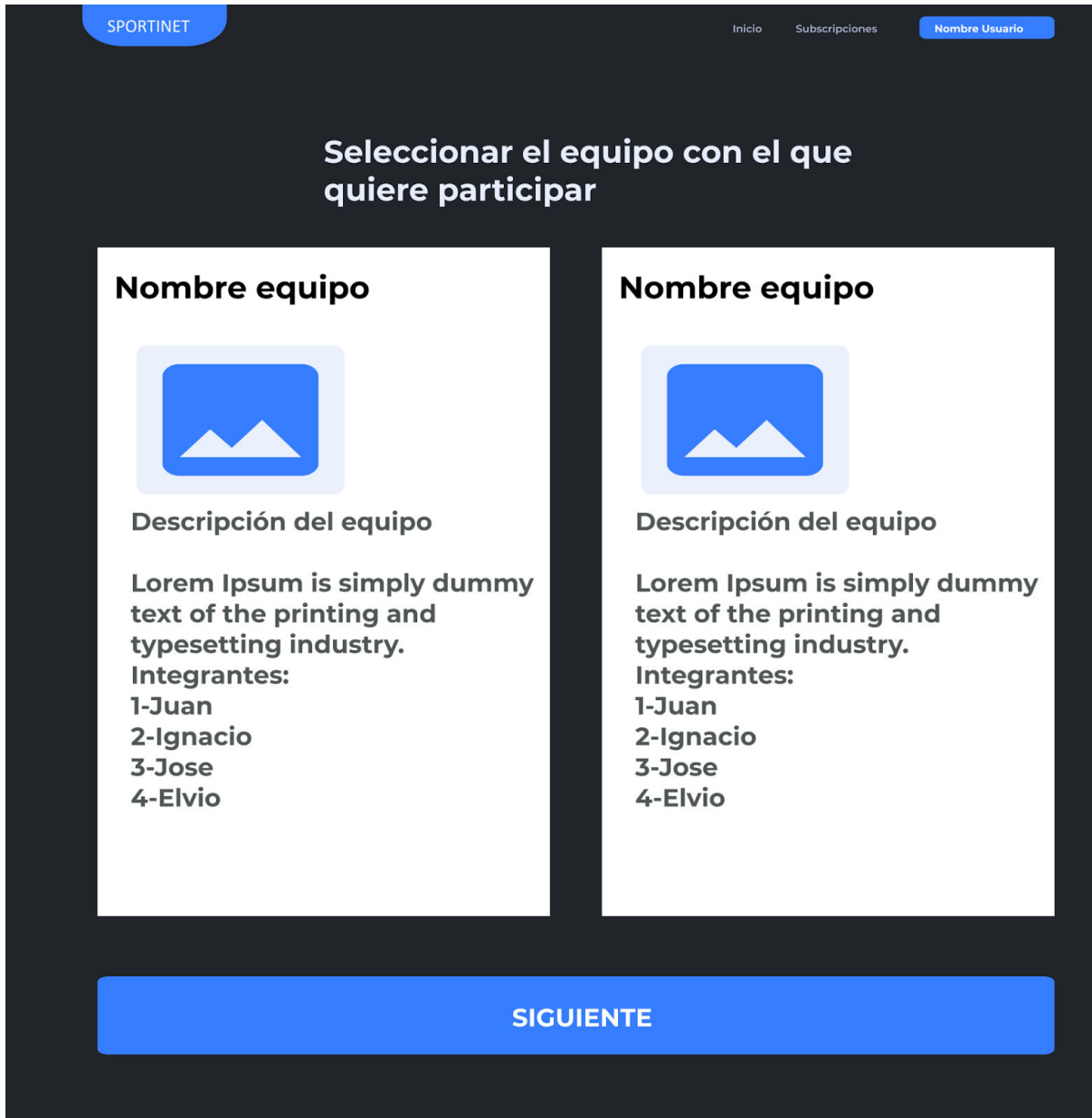


Figura 88: Pantalla de selección de equipo con el que se quiere inscribir al evento

The screenshot displays the 'Pagar con MercadoPago' interface. At the top, there is a navigation bar with 'SPORTINET' on the left, 'Inicio' and 'Subscripciones' in the center, and a user profile button labeled 'Nombre Usuario' on the right. The main heading is 'Pagar con MercadoPago'. Below this, the interface is divided into two main sections: 'Para terminar, confirmá tu pago' and 'Mi tienda'. The 'Para terminar, confirmá tu pago' section includes a user profile for 'Valentina Aguirre' with a 'Cerrar sesión' link, a payment method 'Mastercard \*\*\*\*1234' from 'Banco BBVA' with a 'Modificar' link, and a section titled 'Y completá estos datos' containing a 'Cuotas' dropdown menu set to 'Elegí' and a 'Código de seguridad' field with a note 'Últimos 3 números en el dorso.' and a card image showing the security code. The 'Mi tienda' section shows 'Detalles de tu compra' with 'Productos' for \$ 7.999 and 'Pagás' for \$ 7.999, and a prominent blue 'Pagar' button. At the bottom of the screen, there is a large blue button labeled 'INSCRIBIRSE'.

Figura 89: Pantalla de pago de inscripción

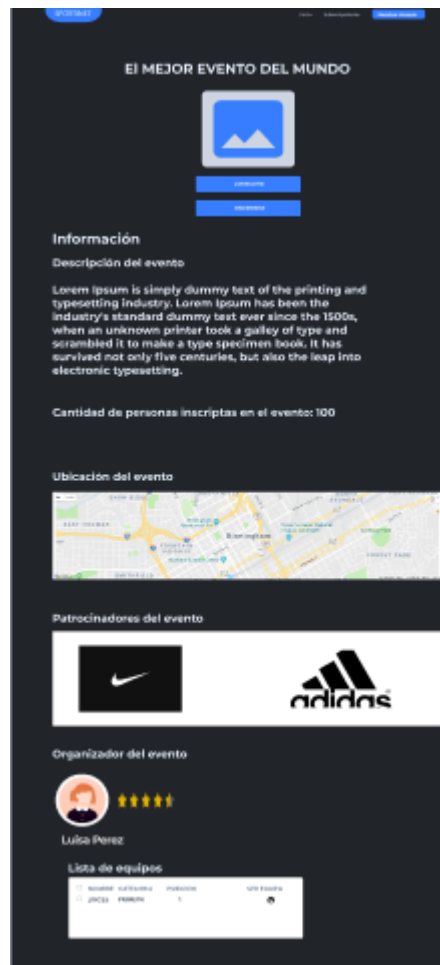


Figura 90: Pantalla para visualizar evento

The screenshot shows a dark-themed web interface for 'SPORTINET'. At the top left, the 'SPORTINET' logo is in a blue rounded rectangle. To its right are links for 'Inicio' and 'Subscripciones', and a blue button labeled 'Nombre Usuario'. The main heading is 'Realizar oferta al organizador del evento'. Below this is a large, light blue rounded rectangular input field with the label 'DESCRIPCIÓN OFERTA:'. At the bottom of the form is a prominent blue button with the text 'ENVIAR SOLICITUD'.

*Figura 91:* Pantalla generación de petición de patrocinio

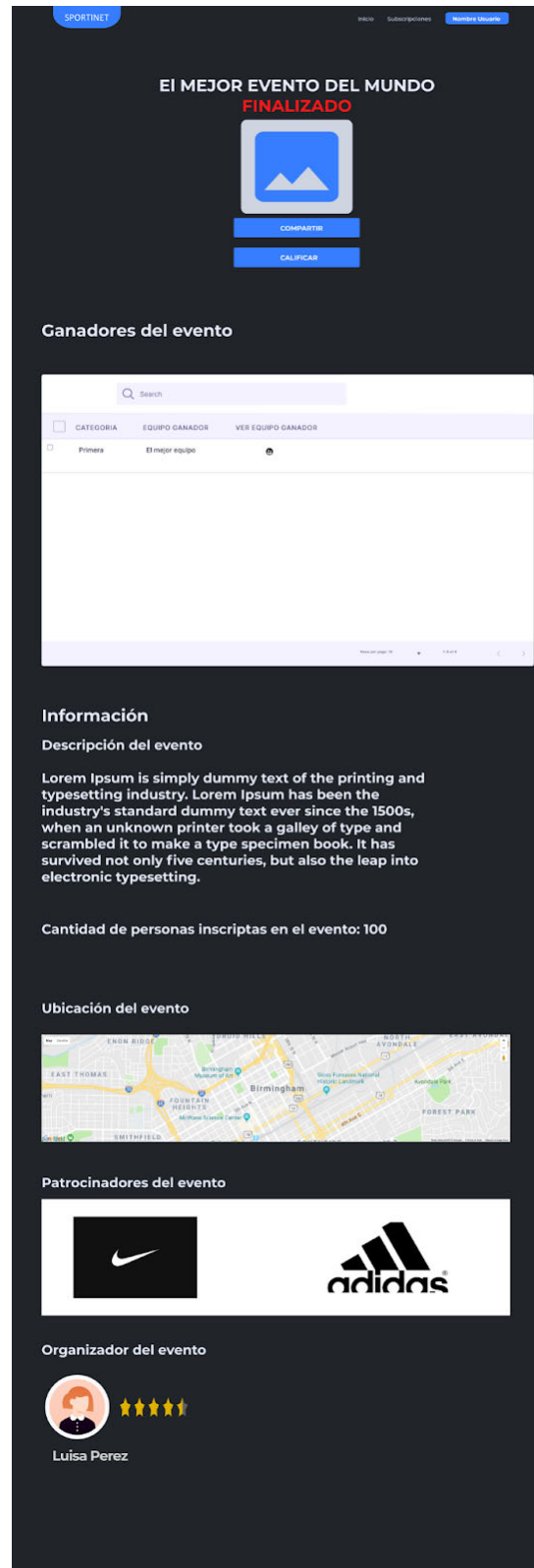


Figura 92: Pantalla de visualización de evento finalizado



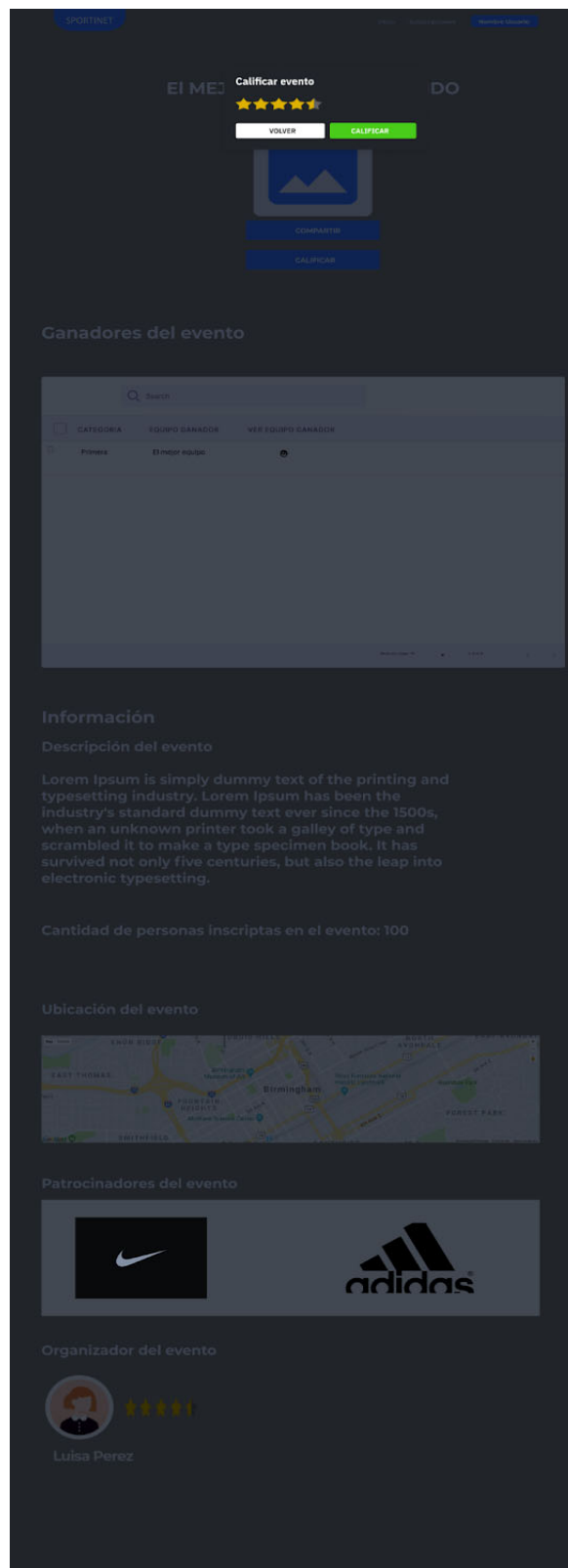


Figura 93: Pantalla de calificación de evento finalizado

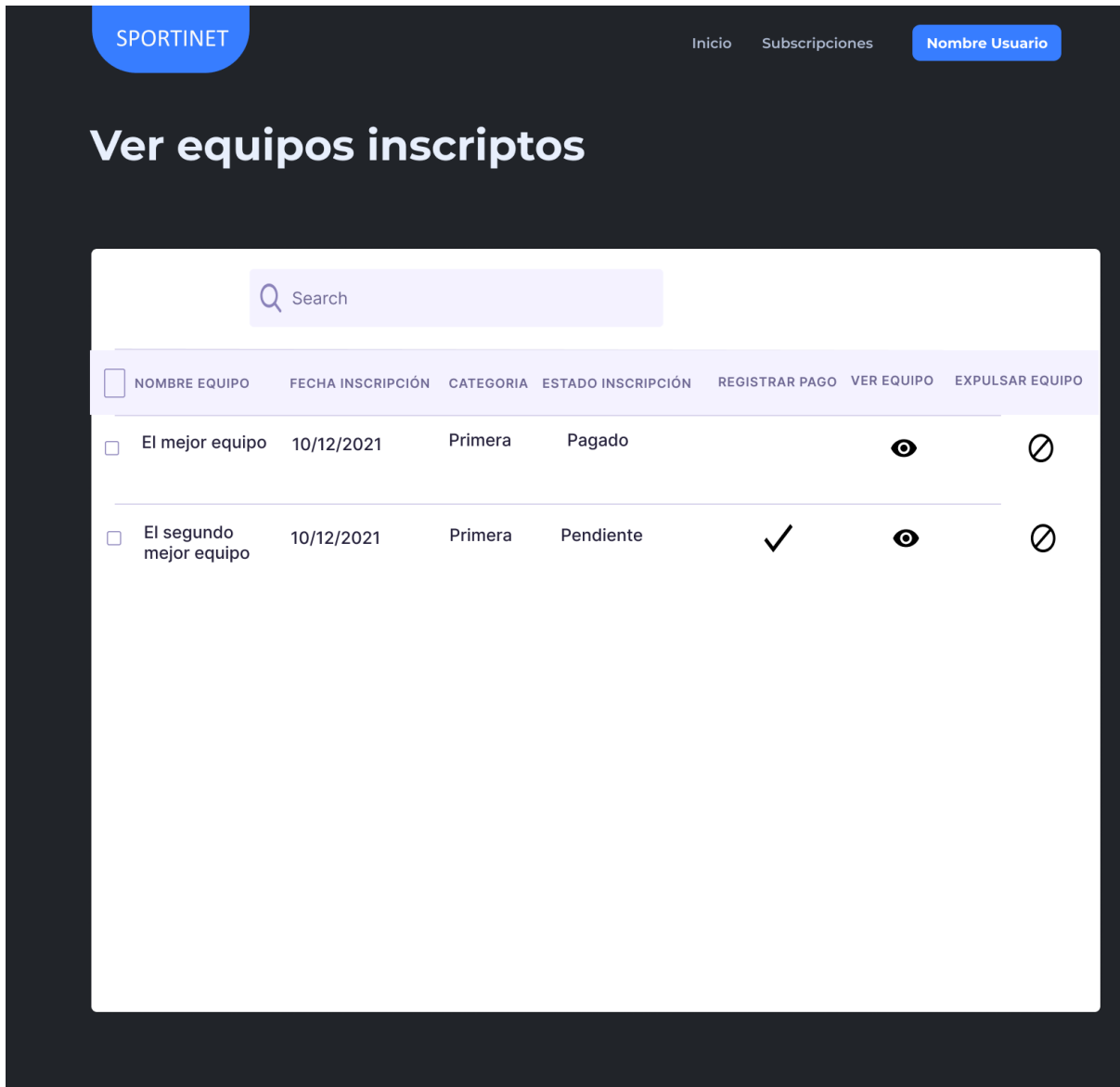


Figura 94: Pantalla de visualización de equipos inscriptos

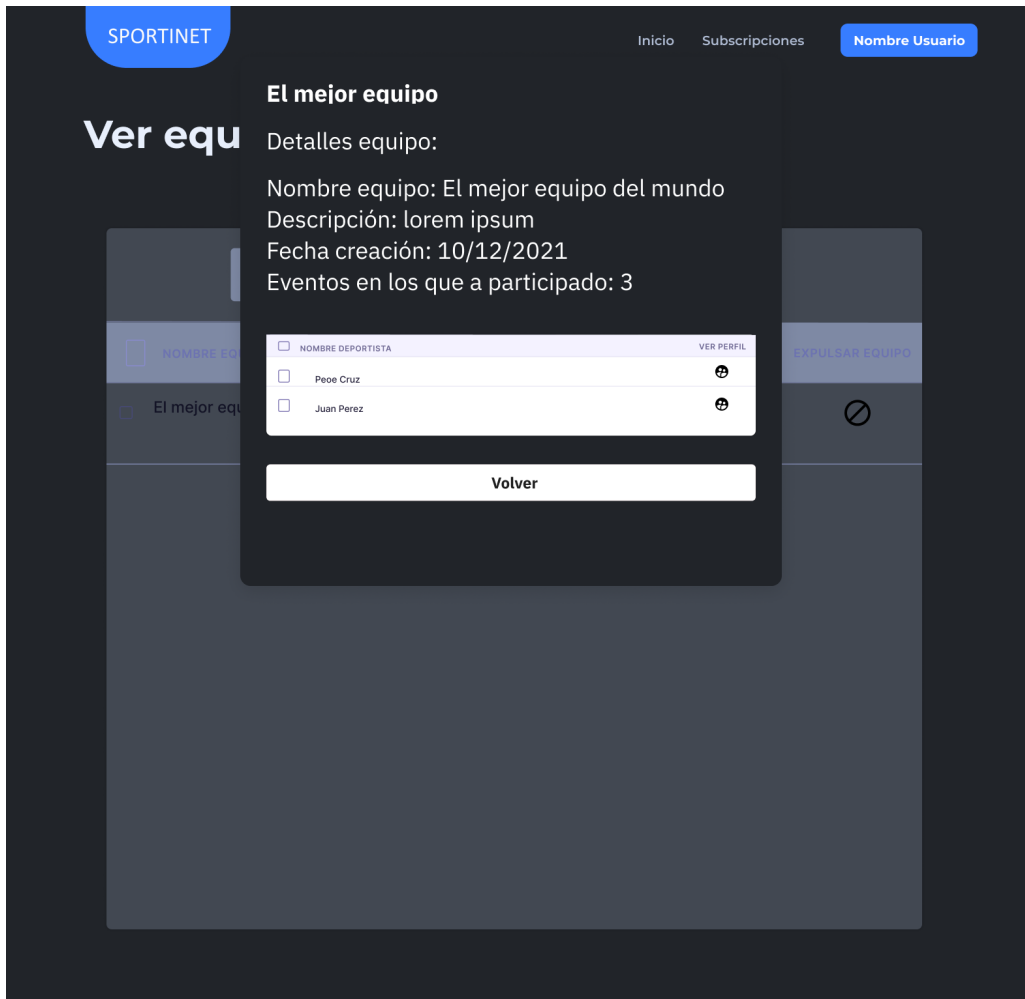


Figura 95: Pantalla de visualización del perfil del equipo

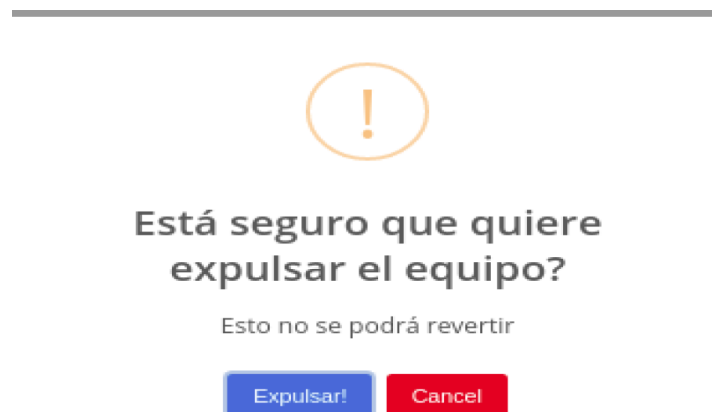


Figura 96: Pantalla de expulsión del equipo



**Está seguro que quiere  
registrar pago del equipo?**

Esto no se podrá revertir

Expulsar!

Cancel


*Figura 97:* Pantalla de registraci3n del pago del equipo



Figura 98: Pantalla de visualización de estadísticas

**SPORTINET** Inicio Subscripciones **Nombre Usuario**

# Luisa Perez



## Descripción

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic

**Tipo usuario: Organizador**  
**Cantidad de eventos organizados: 3**  
**Calificación:**  
★★★★☆

**COMPARTIR**

Figura 99: Pantalla del perfil del usuario

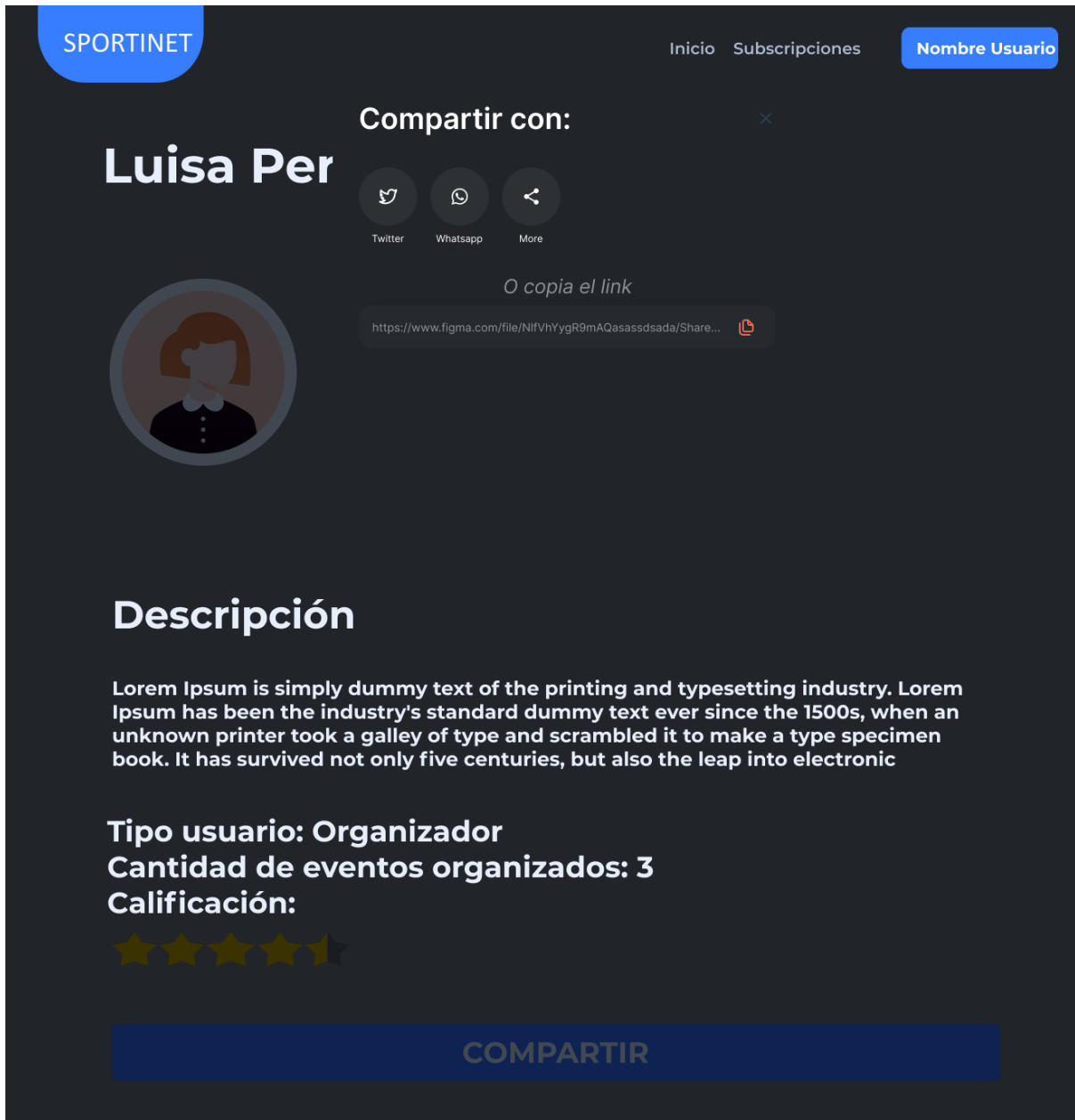


Figura 100: Pantalla para compartir los diferentes perfiles y eventos



**SPORTINEV**

Por favor complete el siguiente cuestionario para que nuestros analistas lo revisen

Ingrese su nombre

Ingrese su apellido

Ingrese su teléfono

Ingrese su email

Mensaje para administrador

Registrarse

Volver

*Figura 101:* Pantalla para contactarse con el administrador



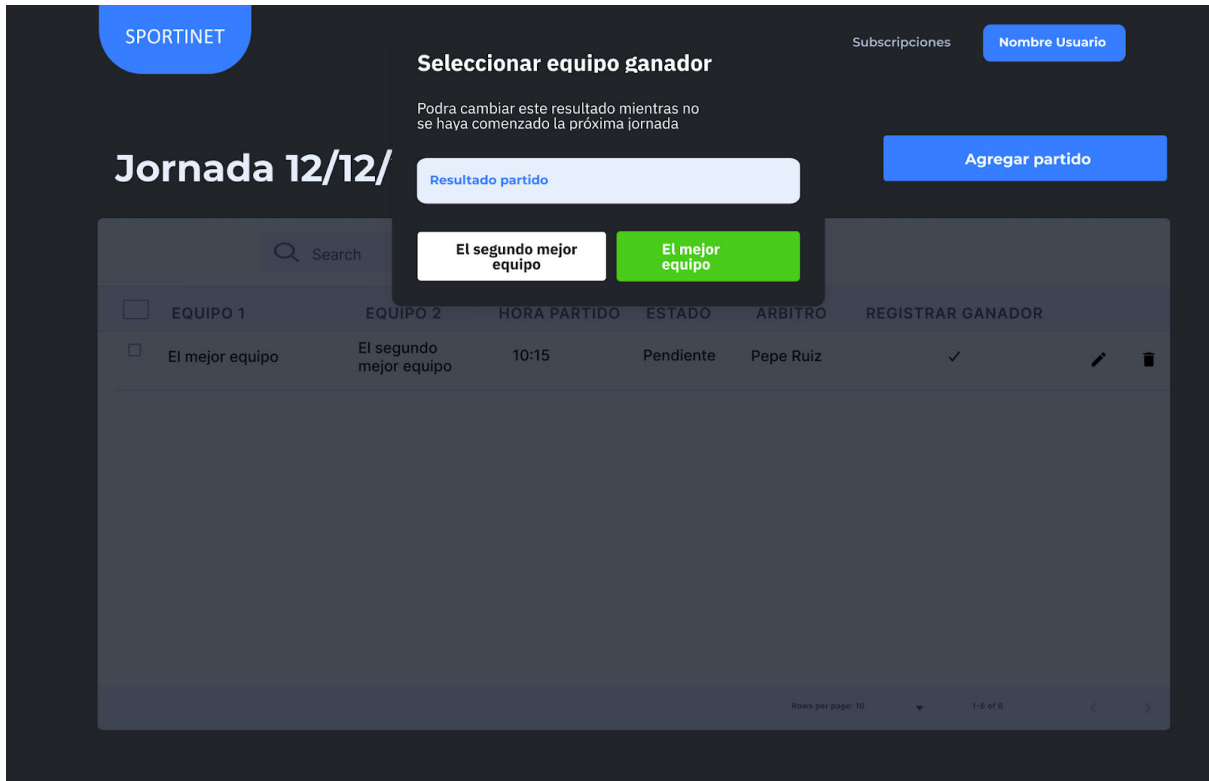


Figura 102: Pantalla de registro de resultado de partido

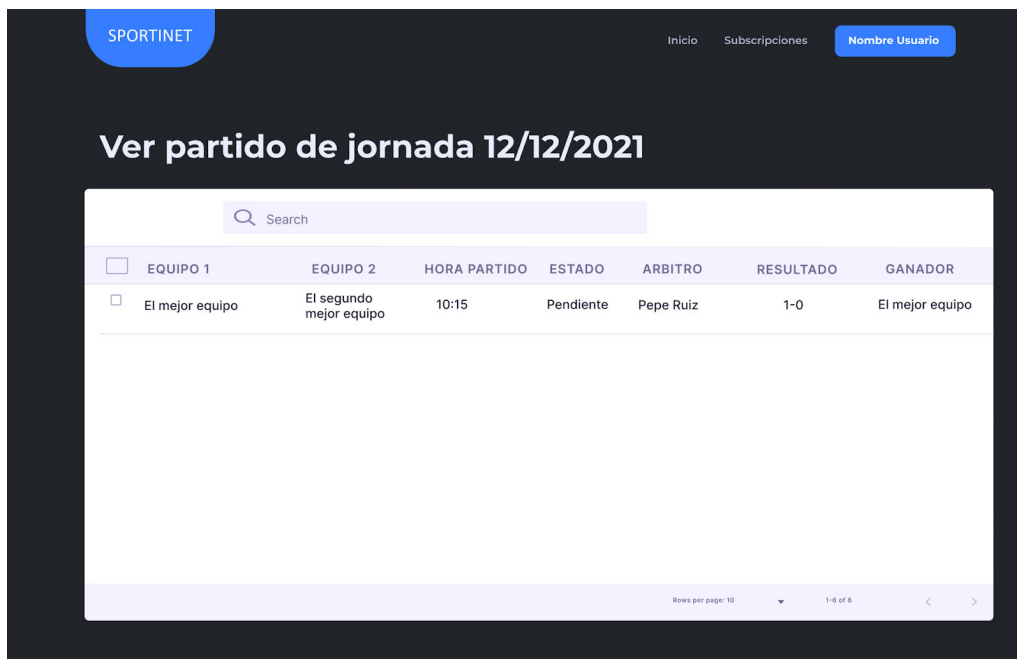


Figura 103: Pantalla de visualización de resultados de la jornada.

The screenshot displays the Sportinev web application interface. At the top, there is a navigation bar with the 'SPORTINEV' logo, 'Inicio', 'Subscripciones', and a 'Nombre Usuario' button. Below the navigation bar is a search bar with the placeholder text 'Buscar evento, organizador, club' and a 'Filtros' button. The main content area is divided into five sections, each displaying a grid of event or organizer cards:

- Eventos cercanos:** A grid of three event cards. Each card displays the event name, organizer name, organizer classification (4.5), event direction, sport (Tennis), and availability date (Abierto hasta 20/09/2020).
- Eventos subscripciones:** A grid of three event cards, identical in layout and content to the 'Eventos cercanos' section.
- Eventos segun tus preferencias:** A grid of three event cards, identical in layout and content to the 'Eventos cercanos' section.
- Mejores organizadores:** A grid of three organizer cards. Each card displays the organizer name, number of tournaments organized (20), average rating (4.5), and number of subscribers (200). A 'Suscribirse' button is located at the bottom right of each card.
- Mejores clubes:** A grid of three club cards. Each card displays the club name, number of tournaments organized (20), average rating (4.5), and number of subscribers (200). A 'Suscribirse' button is located at the bottom right of each card.

Figura 104: Pantalla de visualización de eventos disponibles



Figura 105: Pantalla para establecer la cantidad de publicidad de la aplicación

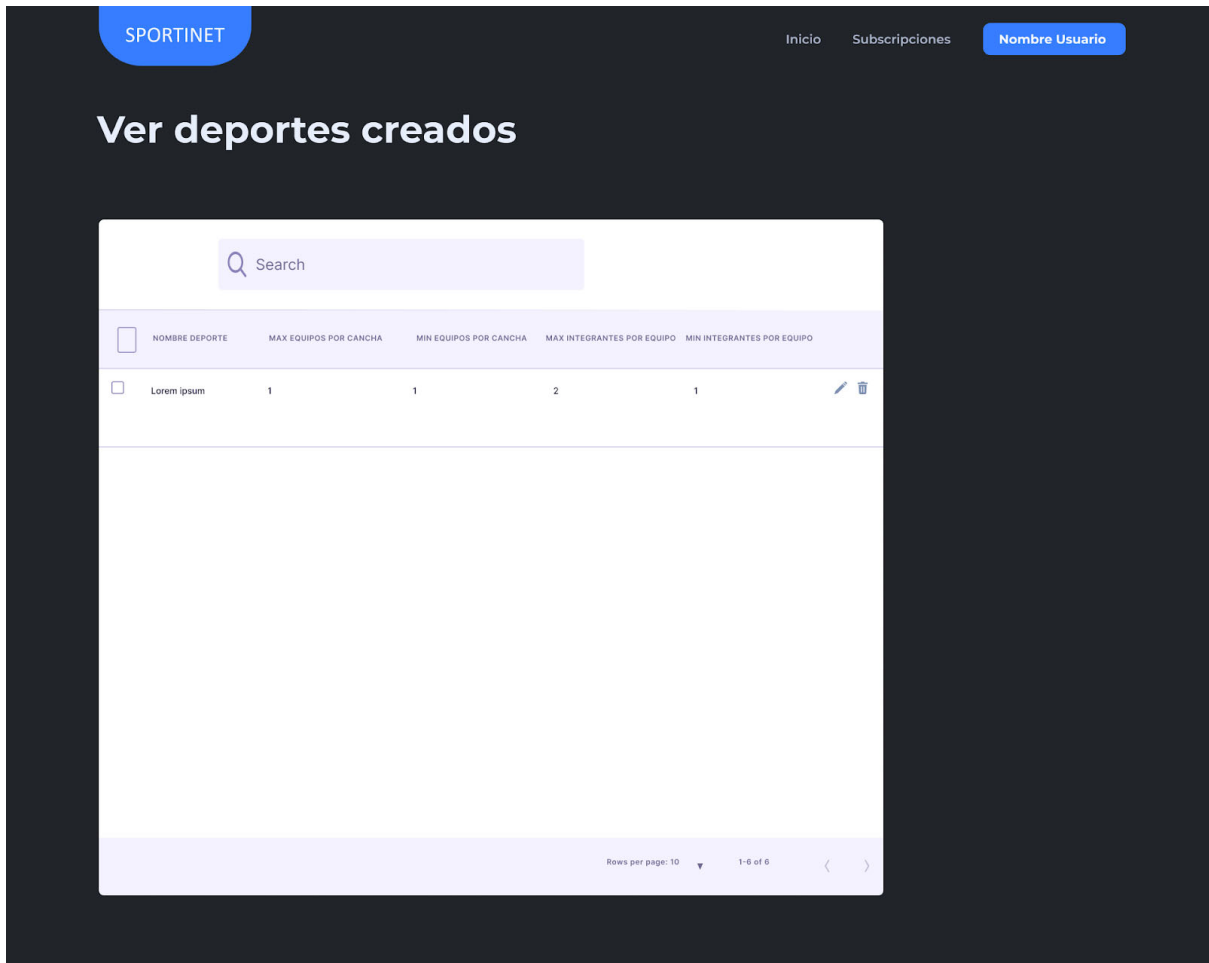


Figura 106: Pantalla de visualización de deportes creados

SPORTINET Inicio Subscripciones Nombre Usuario

## Crear deporte

Nombre deporte

Tipo Eventos

Cantidad de equipos máximos por cancha

Cantidad de equipos mínimos por cancha

Cantidad de integrantes mínimos por equipos

Cantidad de integrantes máximos por equipos

Descripción

Guardar

Figura 107: Pantalla de creación de deporte

The screenshot shows a mobile application interface for 'SPORTINET'. At the top, there is a navigation bar with 'Inicio', 'Subscripciones', and 'Nombre Usuario'. The main heading is 'Modificar deporte'. Below this, there are several input fields: 'Nombre deporte', 'Tipo Eventos', 'Cantidad de equipos máximos por cancha', 'Cantidad de equipos mínimos por cancha', 'Cantidad de integrantes mínimos por equipos', and 'Cantidad de integrantes máximos por equipos'. At the bottom, there is a 'Guardar' button.

Figura 108: Pantalla de modificación de deporte

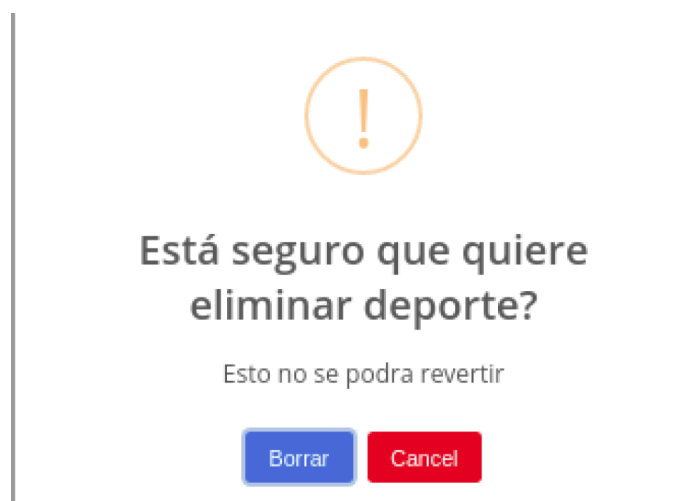


Figura 109: Pantalla de eliminación de deporte

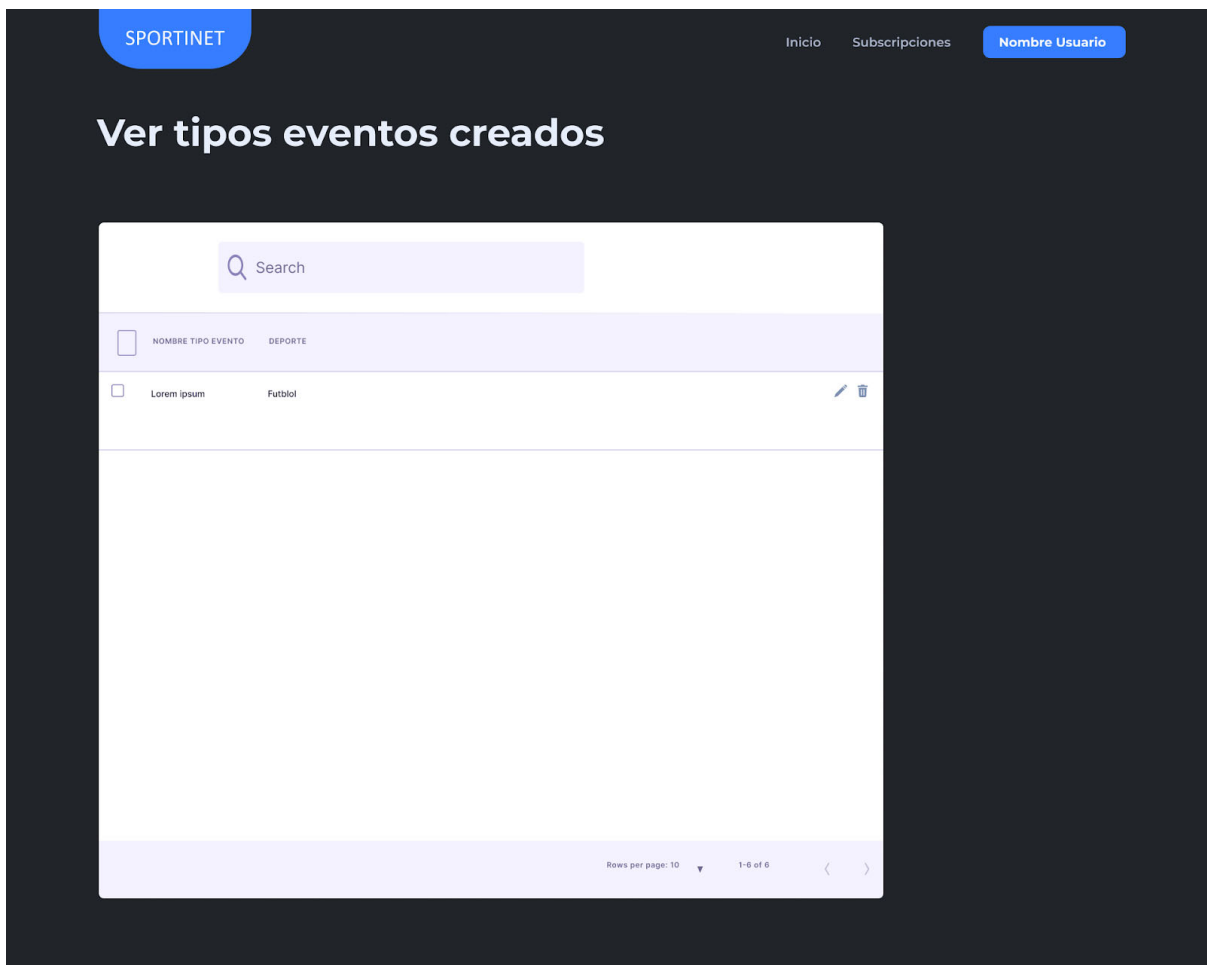


Figura 110: Pantalla de visualización de tipos de eventos creados

SPORTINET

Inicio Subscripciones Nombre Usuario

# Crear tipos eventos

Nombre tipo evento

Deporte

Guardar

Figura 111: Pantalla de creación de tipo evento



SPORTINET

Inicio Subscripciones Nombre Usuario

# Modificar tipos eventos

Nombre tipo evento

Deporte

Guardar

Figura 112: Pantalla de modificación de tipo evento

!

**Está seguro que quiere eliminar tipo evento?**

Esto no se podra revertir

Borrar Cancel

Figura 113: Pantalla de eliminación de tipo evento

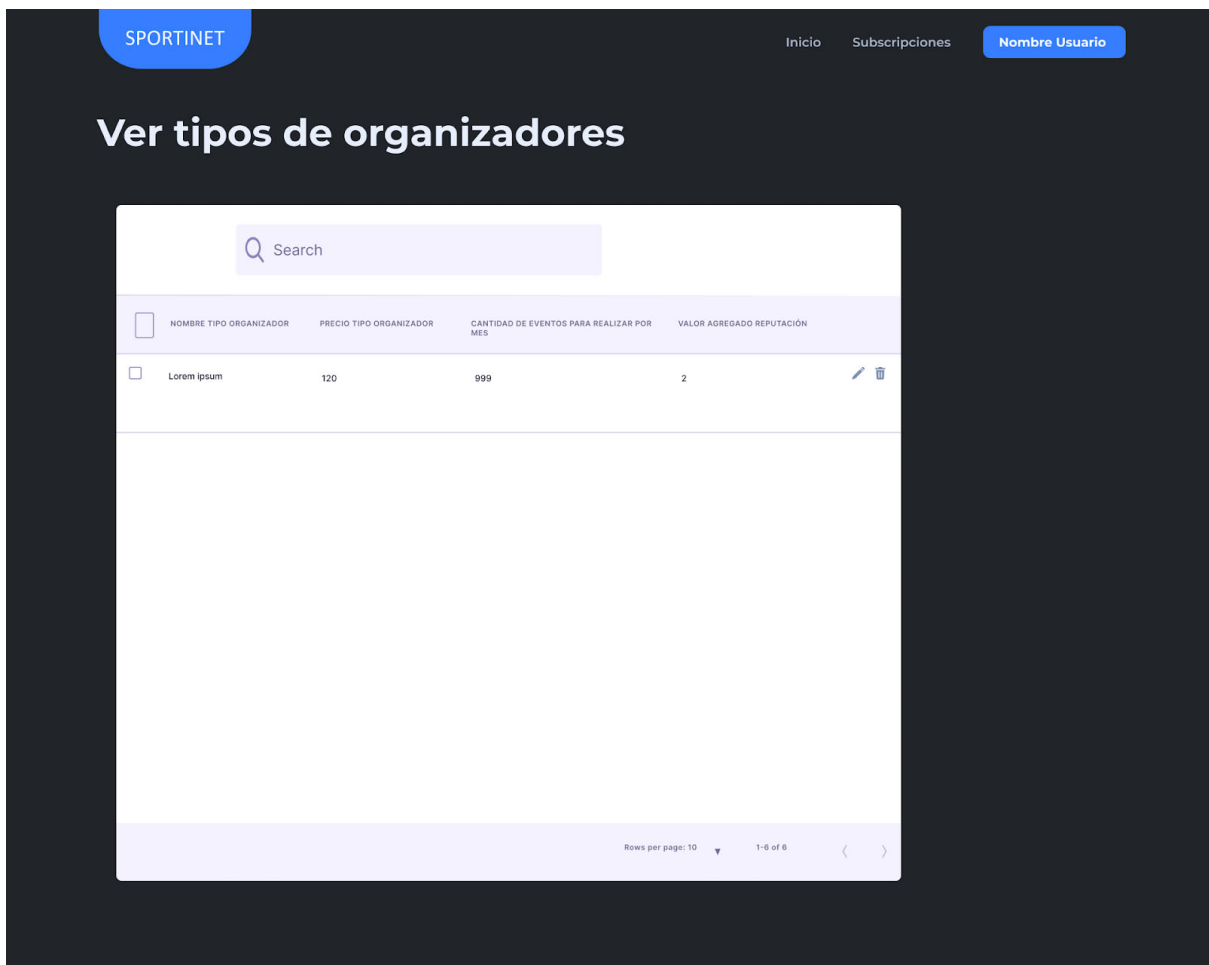


Figura 114: Pantalla de visualización de tipos de organizadores creados

SPORTINET

Inicio Subscripciones Nombre Usuario

## Crear tipos de organizadores

Nombre tipo organizador

Precio tipo organizador

Cantidad de eventos para realizar por mes

Valor agregado para reputación

Guardar

Figura 115: Pantalla de creación de tipo organizador

**SPORTINET** Inicio Subscripciones Nombre Usuario

# Modificar tipos de organizadores

Nombre tipo organizador

Precio tipo organizador

Cantidad de eventos para realizar por mes

Valor agregado para reputación

Guardar

Figura 116: Pantalla de modificación de tipo de organizador

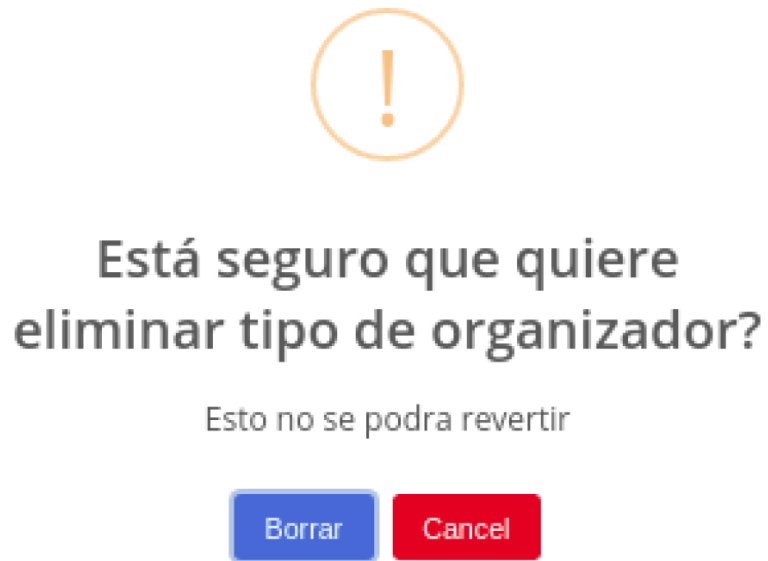


Figura 117: Pantalla de eliminación de tipo organizador

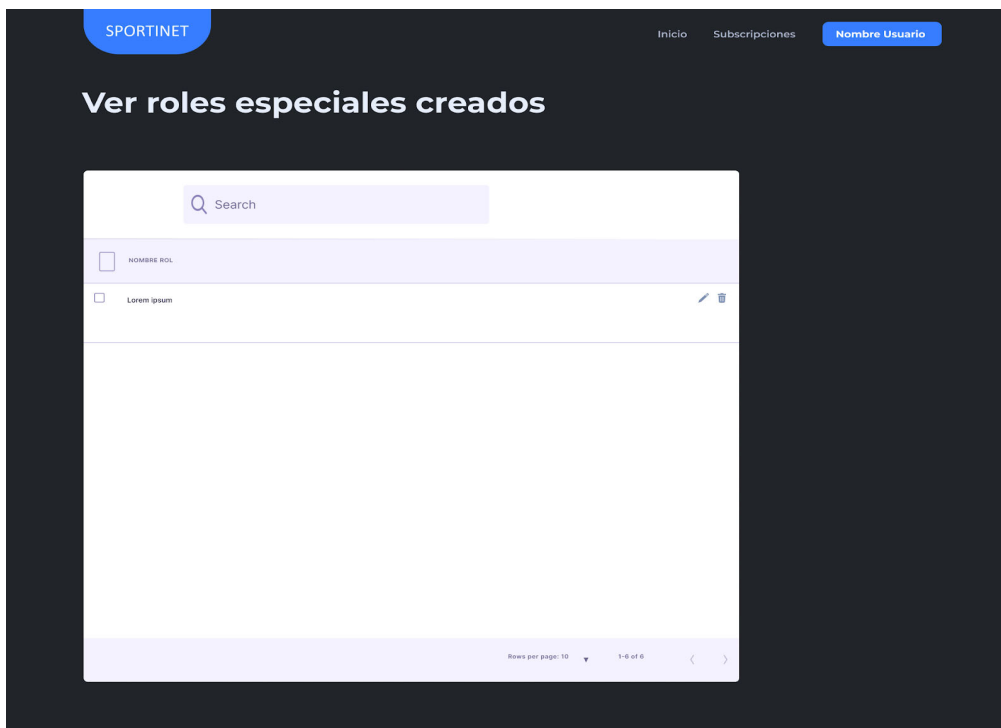


Figura 118: Pantalla de visualización de roles creados

**SPORTINET**

Inicio Subscripciones **Nombre Usuario**

# Crear roles especiales

Nombre rol

Seleccionar permisos

Search

<input type="checkbox"/>	PERMISOS	
<input type="checkbox"/>	Lorem ipsum	

**Guardar**

Figura 119: Pantalla de creación de rol

**SPORTINET** Inicio Subscripciones Nombre Usuario

# Modificar roles especiales

Nombre rol

Seleccionar permisos

Search

<input type="checkbox"/>	PERMISOS	
<input type="checkbox"/>	Lorem ipsum	

Guardar

Figura 120: Pantalla de modificación de rol

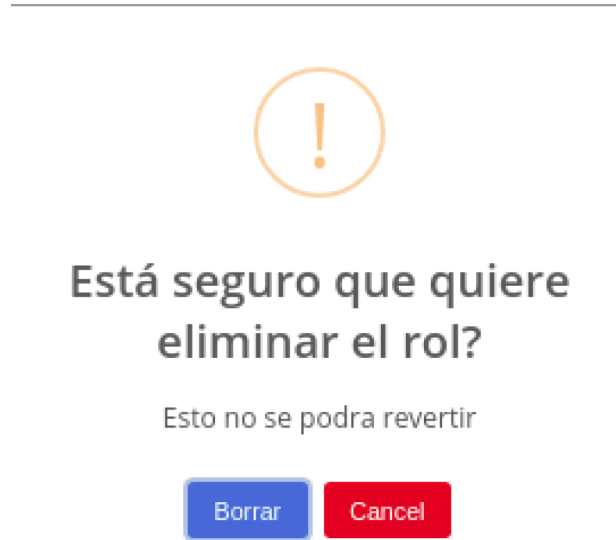


Figura 121: Pantalla de eliminación de rol

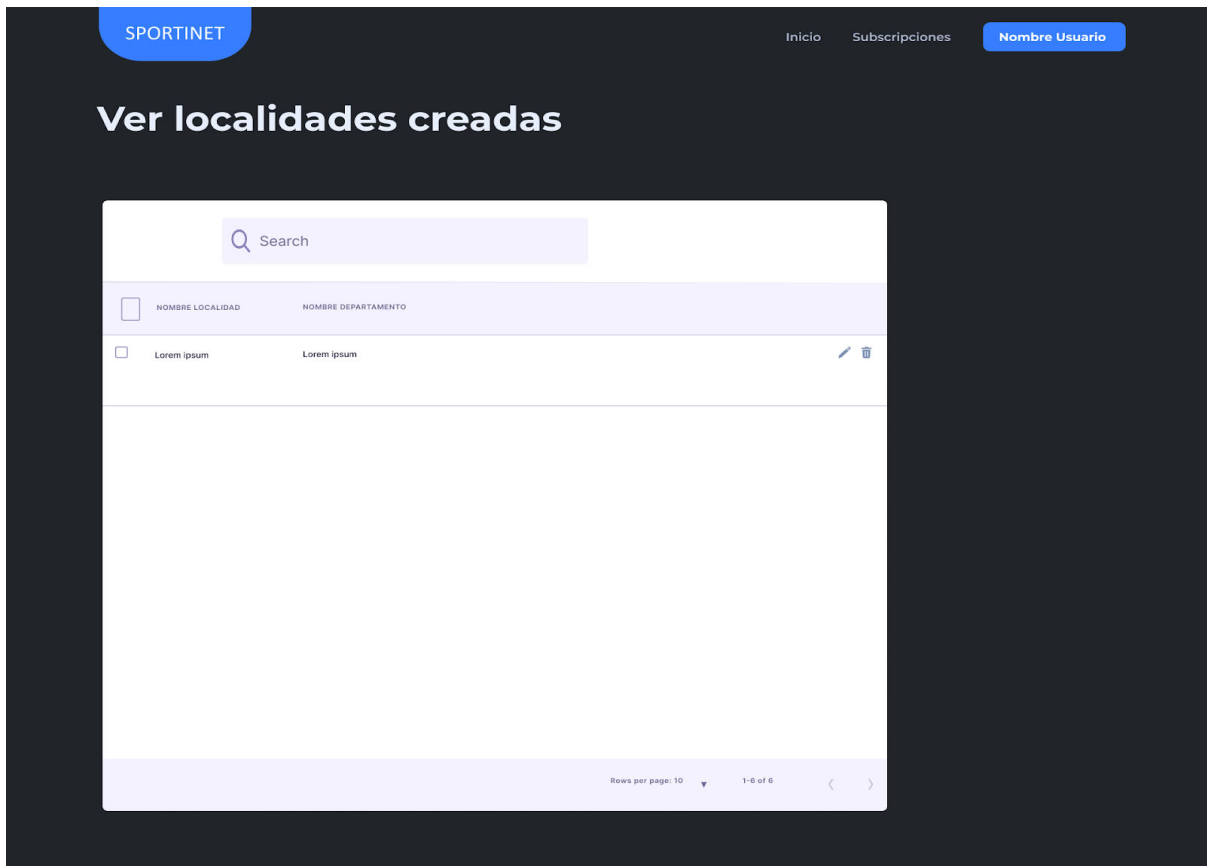




Figura 122: Pantalla de visualización de localidades creadas

SPORTINET

Inicio Subscripciones Nombre Usuario

# Crear localidad

Nombre localidad

Seleccionar departamento

Guardar

Figura 123: Pantalla de creación de localidad

SPORTINET

Inicio Subscripciones Nombre Usuario

# Modificar localidad

Nombre localidad

Seleccionar departamento

Guardar

Figura 124: Pantalla de modificación de localidad



Está seguro que quiere  
eliminar la localidad?

Esto no se podra revertir

Borrar

Cancel

Figura 125: Pantalla de eliminación de localidad

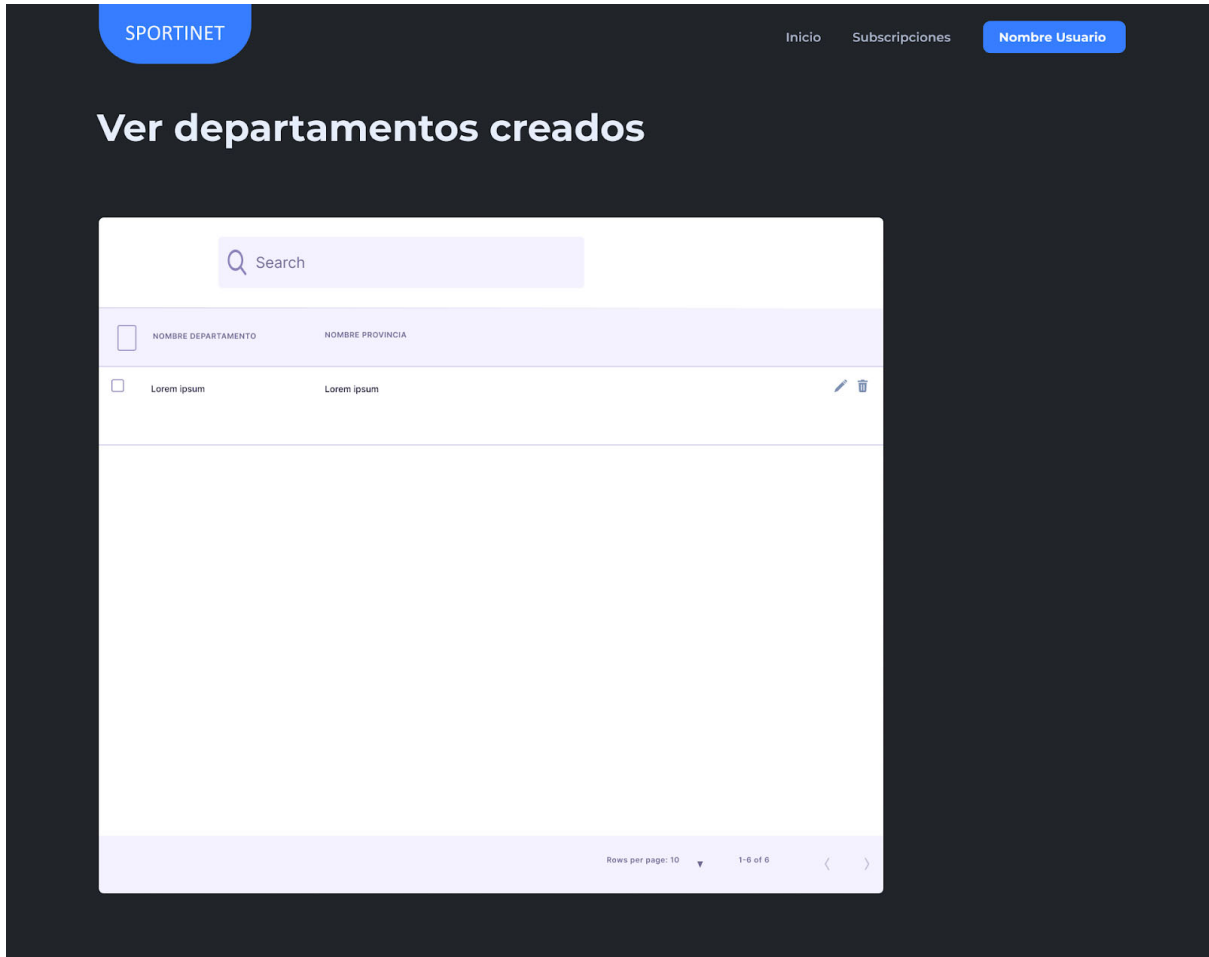



Figura 126: Pantalla de visualización de departamentos creados

**SPORTINET**

Inicio Subscripciones **Nombre Usuario**

# Crear departamento

Nombre departamento

Seleccionar provincia 

**Guardar**

Figura 127: Pantalla creación de departamento

SPORTINET

Inicio Subscripciones Nombre Usuario

# Modificar departamento

Nombre departamento

Seleccionar provincia

Guardar

Figura 128: Pantalla de modificación de departamento



Está seguro que quiere  
eliminar el departamento?

Esto no se podrá revertir

Borrar

Cancel

Figura 129: Pantalla de eliminación de departamento

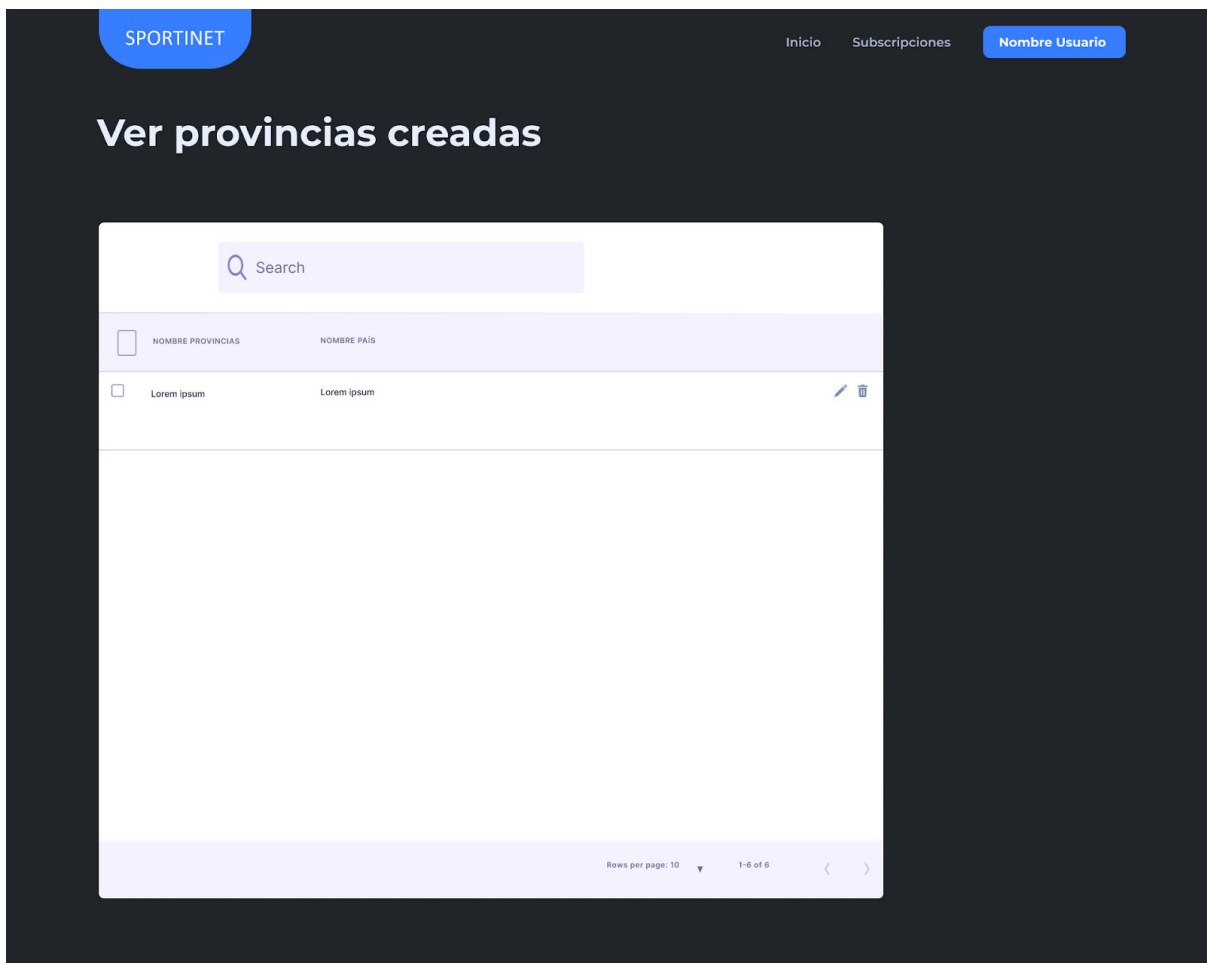


Figura 130: Pantalla de visualización de provincias creadas

SPORTINET

Inicio Subscripciones Nombre Usuario

# Crear provincia

Nombre provincia

Seleccionar país

Guardar

Figura 131: Pantalla de creación de provincia

SPORTINET

Inicio Subscripciones Nombre Usuario

# Modificar provincia

Nombre provincia

Seleccionar país

Guardar

Figura 132: Pantalla de modificación de provincia

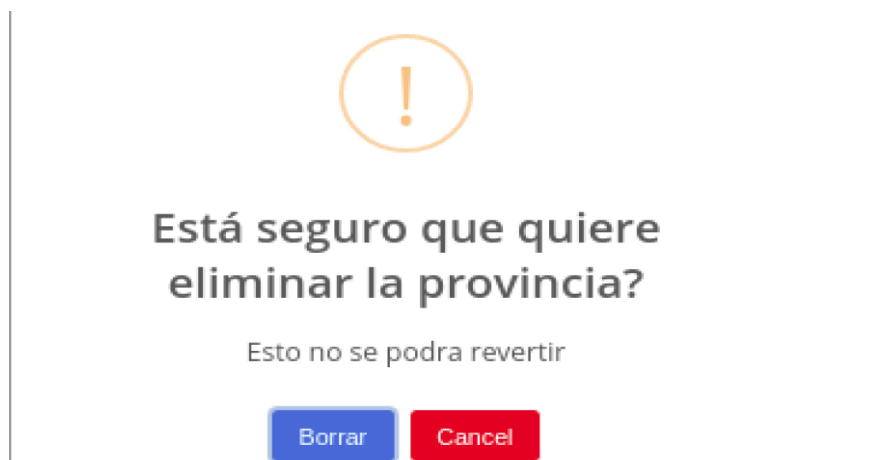




Figura 133: Pantalla de eliminación de provincia

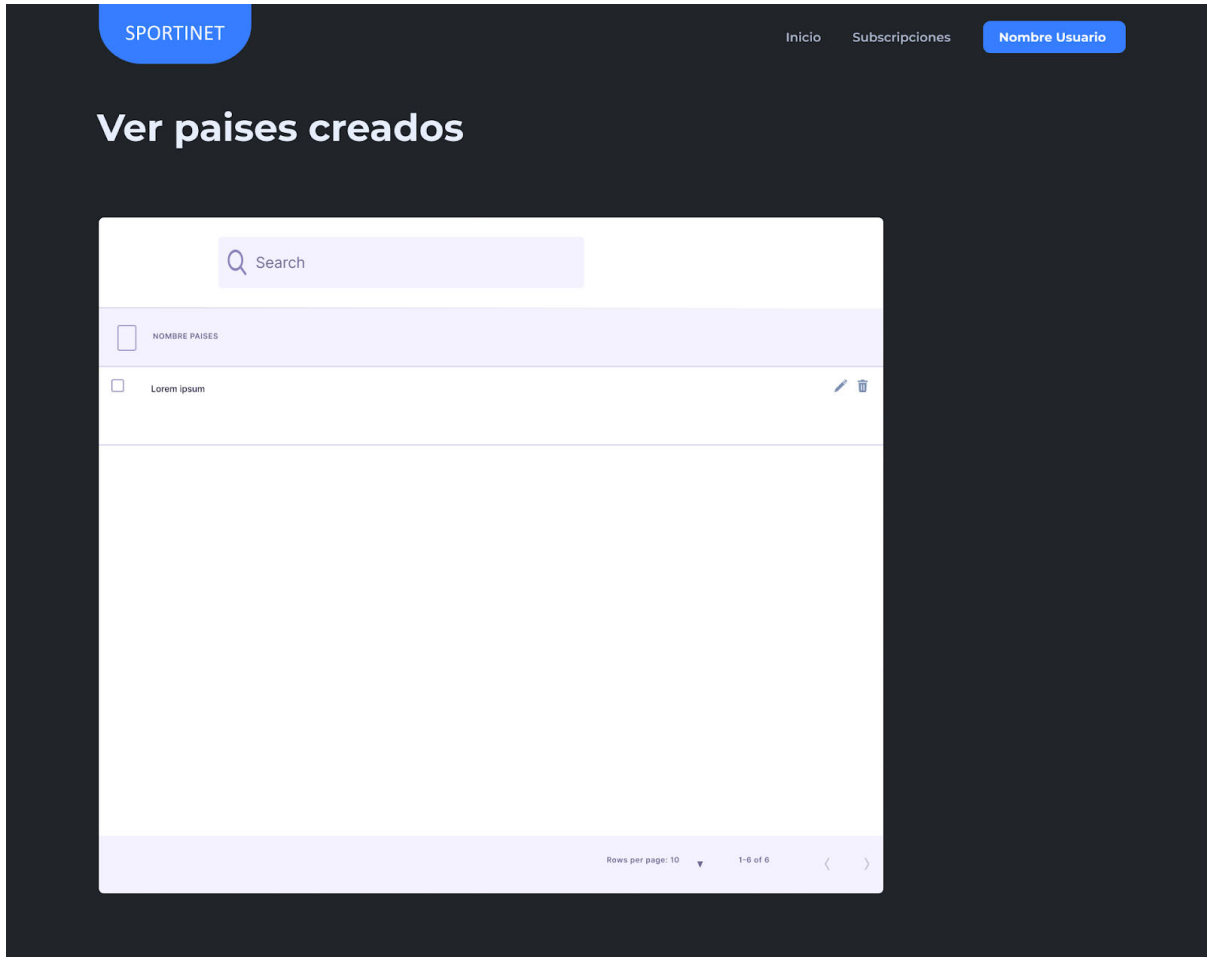


Figura 134: Pantalla de visualización de países creados

The screenshot shows the 'Crear países' (Create countries) screen. At the top left is the 'SPORTINET' logo. To its right are navigation links for 'Inicio' and 'Subscripciones', and a user profile button labeled 'Nombre Usuario'. The main heading is 'Crear países'. Below it is a light blue input field with the placeholder text 'Nombre país'. At the bottom right is a blue button labeled 'Guardar'.

Figura 135: Pantalla de creación de país

The screenshot shows the 'Modificar países' (Modify countries) screen. The layout is identical to the 'Crear países' screen, featuring the 'SPORTINET' logo, navigation links for 'Inicio' and 'Subscripciones', a user profile button labeled 'Nombre Usuario', the main heading 'Modificar países', a light blue input field with the placeholder text 'Nombre país', and a blue button labeled 'Guardar'.

Figura 136: Pantalla de modificación de país



Está seguro que quiere  
eliminar el país?

Esto no se podrá revertir

Borrar

Cancel

Figura 137: Pantalla de eliminación de país

SPORTINET Inicio Subscripciones Nombre Usuario

## Ver canchas creadas

Search




NOMBRE CANCHA	NUMERO CANCHA	PRECIO	ESTADO	DEPORTES
Cancha 1	1	120	Disponible	  

Figura 138: Pantalla de visualización de canchas creadas

SPORTINEV Inicio Subscripciones Nombre Usuario

## Crear cancha

Nombre cancha

Número cancha

Precio cancha por día

Estado cancha

Descripción estado cancha

Capacidad

Seleccionar deportes

Search

<input type="checkbox"/>	NOMBRE DEPORTE	
<input type="checkbox"/>	Futbol	
<input type="checkbox"/>	Tenis	

Guardar

Figura 139: Pantalla de creación de canchas

SPORTINEV Inicio Subscripciones Nombre Usuario

## Modificar cancha

Nombre cancha

Número cancha

Precio cancha por día

Estado cancha

Descripción estado cancha

Capacidad

Seleccionar deportes

Buscar

NOMBRE DEPORTES

Fútbol

Tenis

Guardar

Figura 140: Pantalla de modificación de cancha



Está seguro que quiere  
eliminar la cancha?

Esto no se podrá revertir!

Borrar

Cancel

Figura 141: Pantalla de eliminación de cancha

SPORTINET Inicio Subscripciones Nombre Usuario

## Cambiar Email

Nuevo Mail

Contraseña

Guardar

Figura 142: Pantalla para cambiar email

SPORTINET Inicio Subscripciones Nombre Usuario

## Cambiar Contraseña

Nueva Contraseña

Confirmar Contraseña

Contraseña Anterior

Guardar

Figura 143: Pantalla para cambiar contraseña

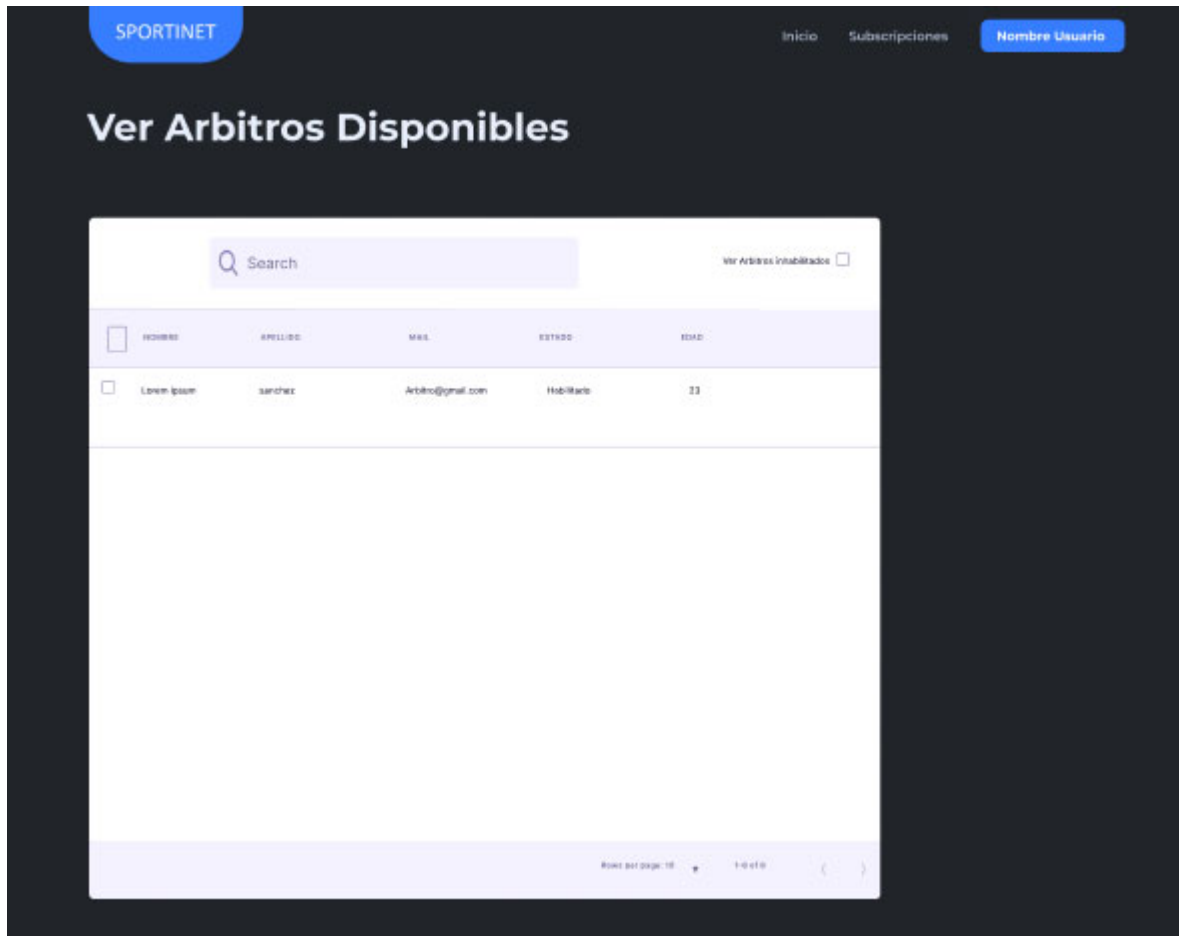


Figura 144: Pantalla de visualización de árbitros



The screenshot shows the 'Crear Jornada' (Create Shift) screen. At the top, there is a navigation bar with 'SPORTINET' on the left, 'Inicio' and 'Subscripciones' in the center, and a 'Nombre Usuario' button on the right. The main title 'Crear Jornada' is prominently displayed. Below the title, there are three input fields: 'Fecha Inicio' (Start Date), 'Hora Desde' (Start Time), and 'Hora Hasta' (End Time). Each field has a dropdown arrow on its right side. At the bottom of the form, there is a large blue 'Guardar' (Save) button.

Figura 145: Pantalla de Creación de Jornadas

The screenshot shows the 'Crear Partido' (Create Match) screen. It features the same top navigation bar as the previous screen. The main title is 'Crear Partido'. Below the title, there are four input fields: 'Cancha' (Field), 'Arbitro' (Referee), 'Fecha Inicio' (Start Date), and two time fields 'Hora Desde' (Start Time) and 'Hora Hasta' (End Time). Below these fields, there is a section titled 'Añadir equipos' (Add teams) with an 'Agregar Equipo' (Add Team) button. Underneath, there is a table with two columns: 'NOMBRE' (Name) and 'ESTADO' (Status). The first row shows 'chilotes' and 'Habilitado'. To the right of the table, there is a trash icon labeled 'ELIMINAR EQUIPO' (Delete Team). At the bottom, there is a large blue 'Guardar' (Save) button.

Figura 146: Pantalla de Creación de Partido

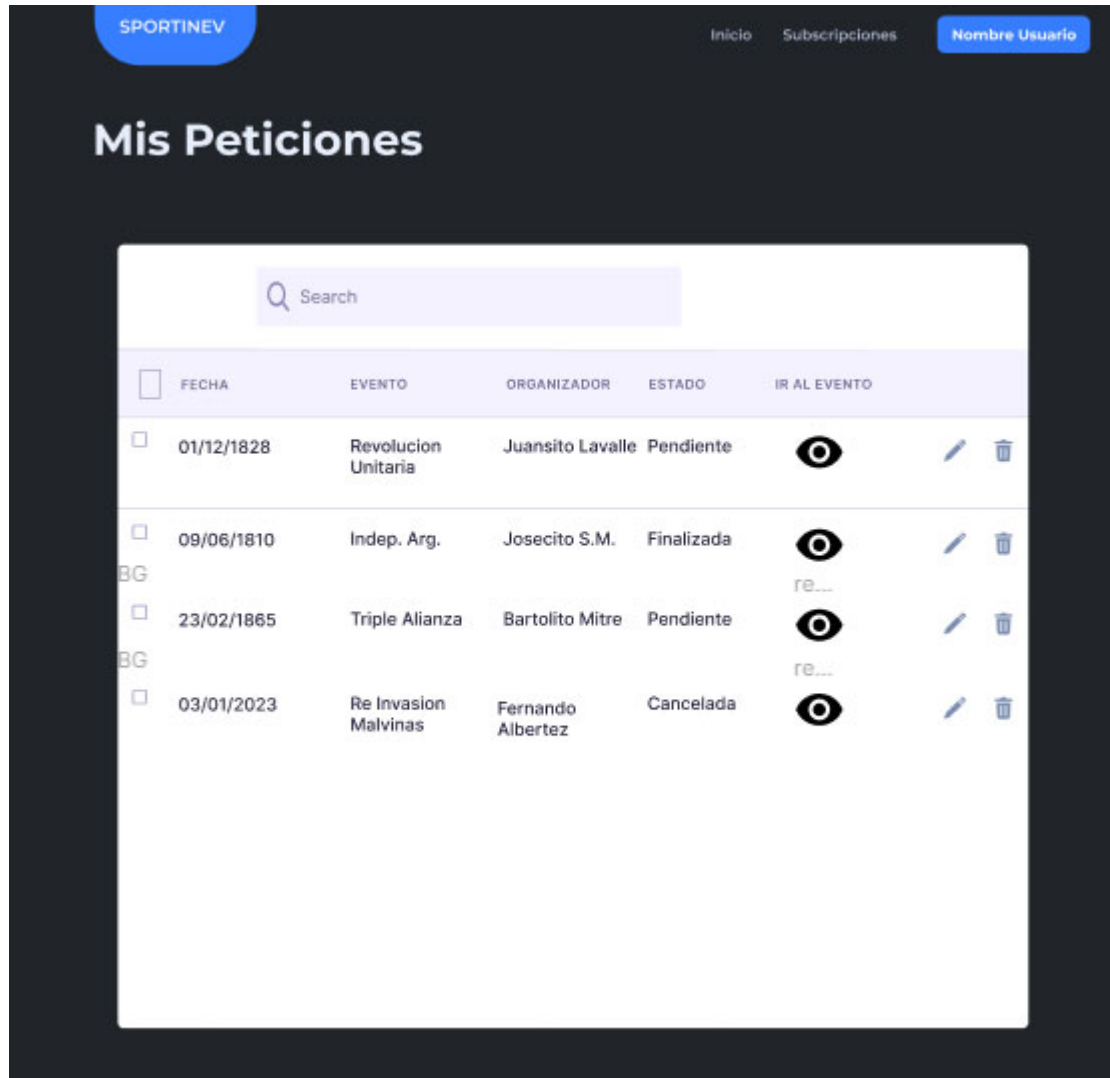


Figura 147: Pantalla de Visualización de peticiones de patrocinio como patrocinador



Figura 148: Pantalla de Visualización de Eventos Patrocinados Como Patrocinador

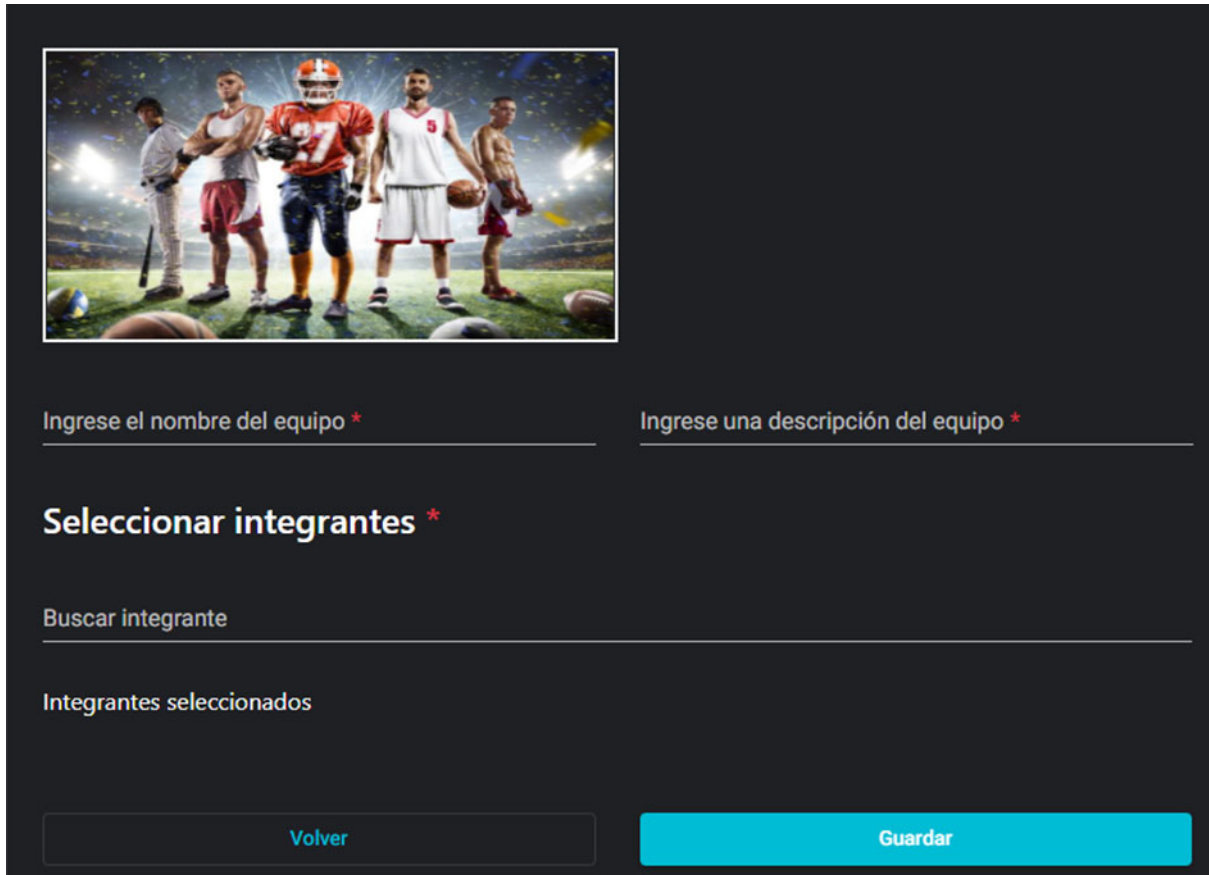
Mis equipos Agregar

Filtro

Nombre	Descripción	Club	Integrantes	Compartir	Salir de equipo	Actions
Messsiiii	esy	messi				
equipo de Messi	solo messi					
linces	as					
linces 2	asdasd					
un equipo ft	asd					

Items por página: 10 ▼ 1 - 9 of 9 < >

Figura 149: Pantalla de Visualización de Equipos como deportista / club



Ingrese el nombre del equipo \*

Ingrese una descripción del equipo \*

**Seleccionar integrantes \***

Buscar integrante

Integrantes seleccionados

Volver

Guardar

Figura 150: Pantalla de alta y modificación de equipos

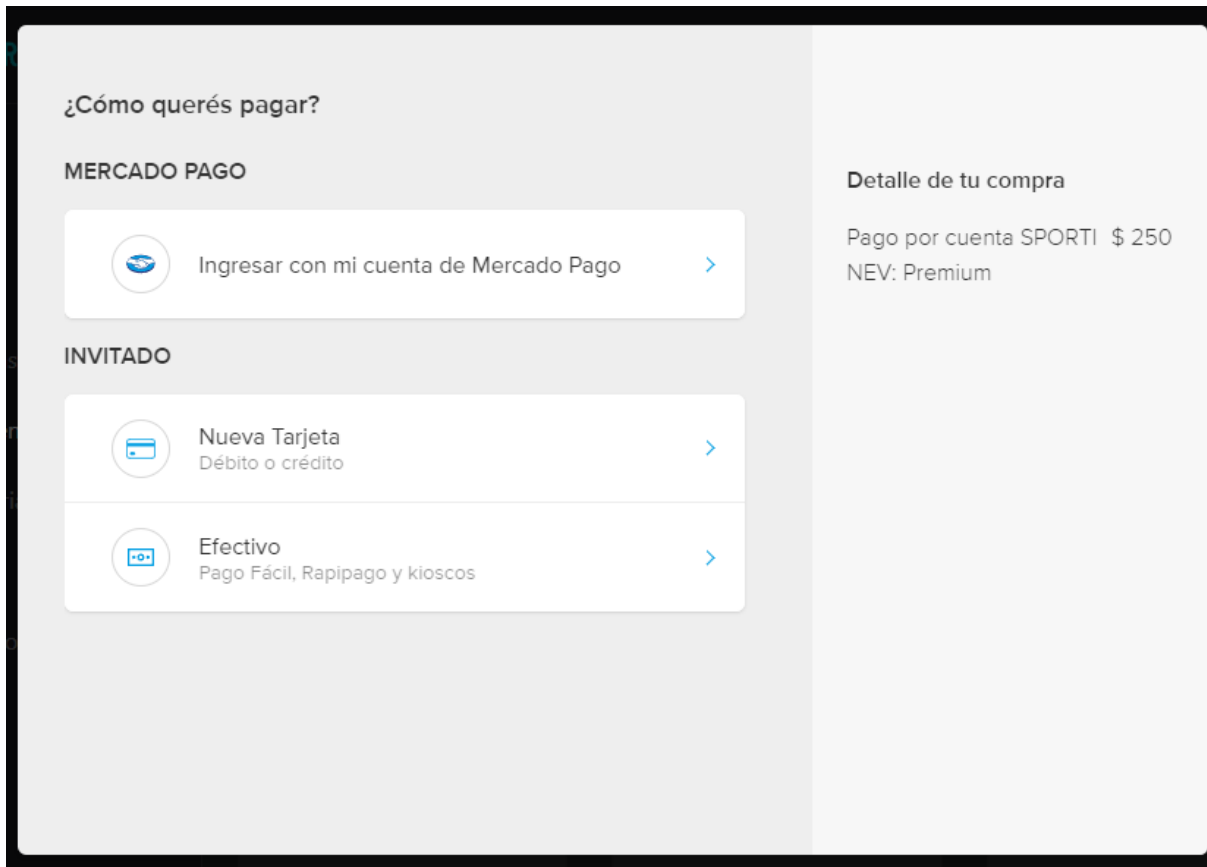


Figura 151: Pantalla de visualización de la integración con Mercado Pago

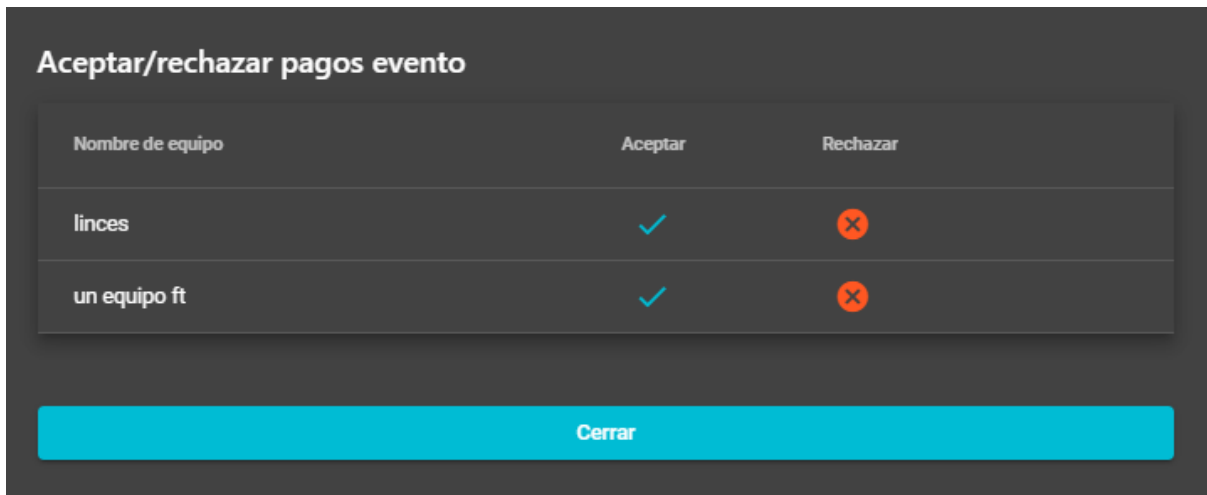


Figura 152: Ventana de registro de pagos de equipos en un evento

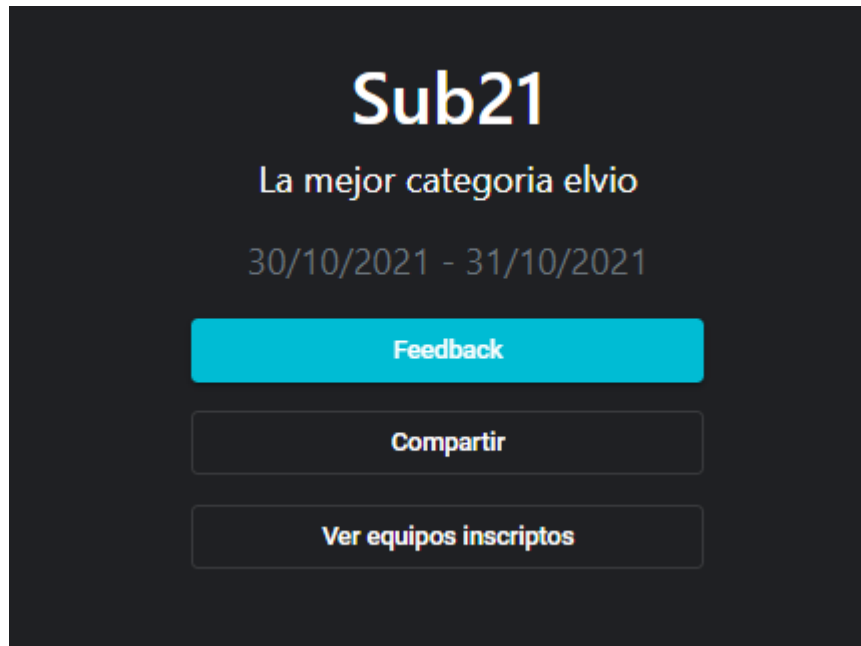


Figura 153: Botón de feedback en categoría de evento

Contanos que te parecio el evento...

el evento estuvo **boenisimo**, me **gustaria** dejar de ser tan bueno

★★★★★

### Feedback del club: River Plate

Contanos que te parecio el club...









el mejor club ever

★★★★★

Figura 154: Ventana de feedback

Mis categorías Agregar

Filtro

Nombre	Observaciones	Deporte	Edad mínima	Edad máxima	Peso mínimo	Peso máximo	Actions
Primera	La mejor categoría del mundo						 
Primera tenis			16	45			 
Reserva	Categoría de la reserva		16	26	70	100	 
Sub 15			12	15			 

Items por página: 10 1 - 4 of 4

Figura 155: Pantalla de visualización de las categorías.

### Categoría

Ingrese el nombre de la categoría \*

Profesional

---

Ingrese observaciones

Categoría para los profesionales

---

Edad mínima de la categoría \*

16

Edad máxima de la categoría \*

45

Peso mínimo de la categorías [kg]

Peso máximo de la categoría [kg]

### Seleccionar deporte

Buscar...

Futbol

---

Deporte seleccionado

Futbol

---

Volver

Guardar

Figura 156: Pantalla de modificación de las categorías.

### Deportes favoritos

Buscar deportes

Futbol-Tenis

Futbol-Tenis

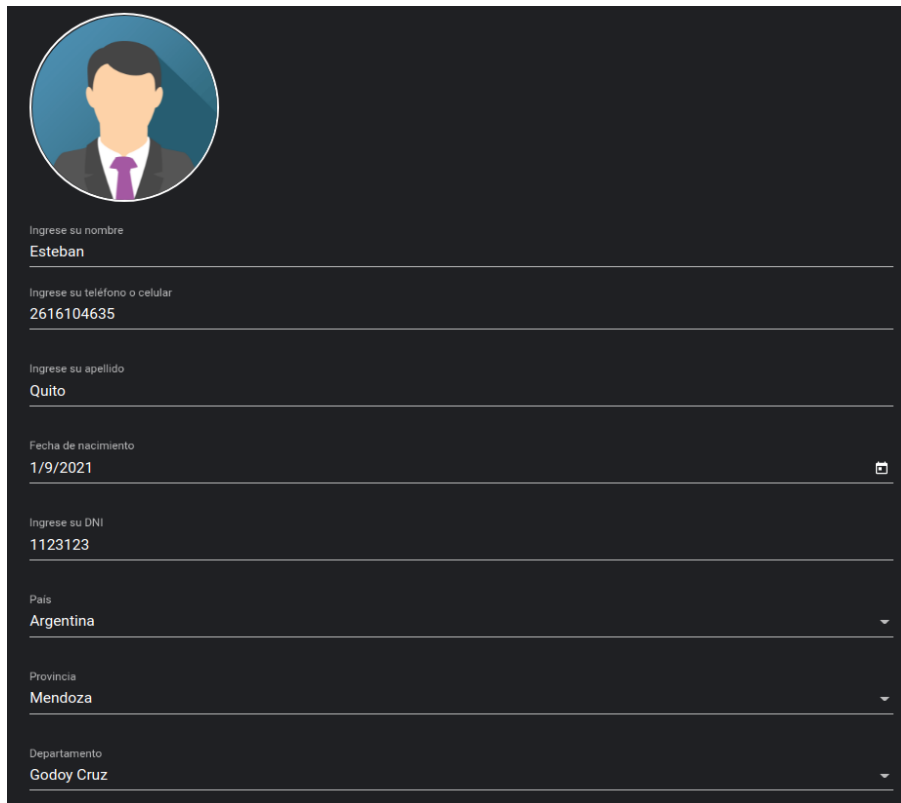
Futbol × Futbol-Tenis ×

Cancelar

Guardar cambios

Figura 157: Creación de deporte favorito

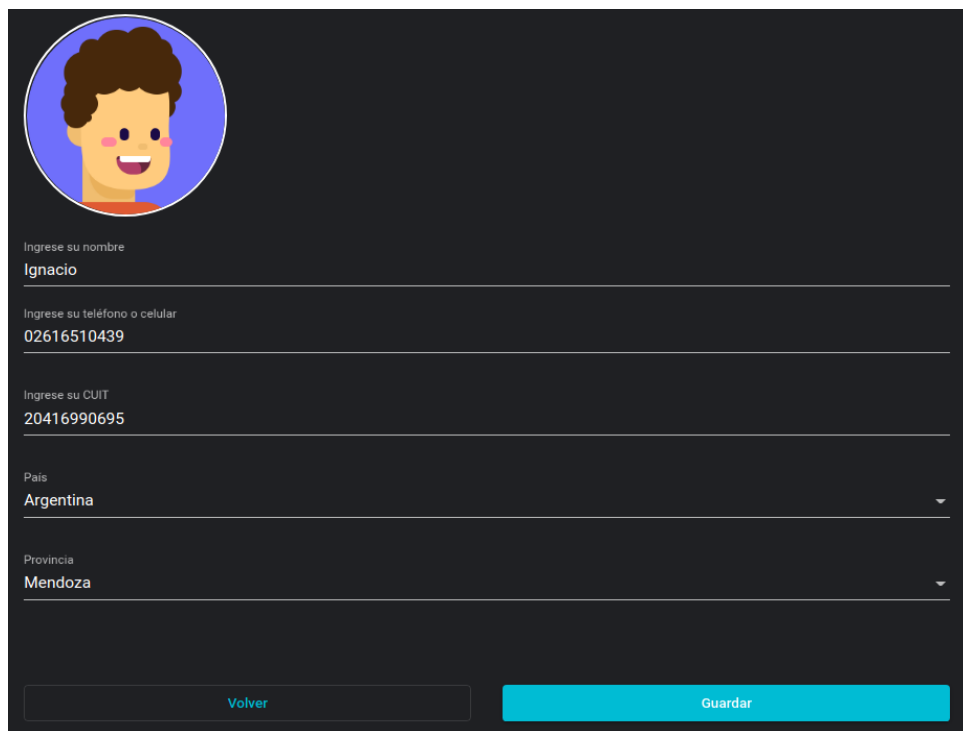




A screenshot of a user profile editing form on a dark background. At the top left is a circular profile picture of a man in a suit. Below it are several input fields with labels and values:

- Ingrese su nombre:** Esteban
- Ingrese su teléfono o celular:** 2616104635
- Ingrese su apellido:** Quito
- Fecha de nacimiento:** 1/9/2021 (with a calendar icon)
- Ingrese su DNI:** 1123123
- Pais:** Argentina (dropdown menu)
- Provincia:** Mendoza (dropdown menu)
- Departamento:** Godoy Cruz (dropdown menu)

Figura 158: Pantalla que nos muestra como editar el perfil desde la perspectiva del deportista.




A screenshot of a user profile editing form on a dark background. At the top left is a circular profile picture of a smiling man with curly hair. Below it are several input fields with labels and values:

- Ingrese su nombre:** Ignacio
- Ingrese su teléfono o celular:** 02616510439
- Ingrese su CUIT:** 20416990695
- Pais:** Argentina (dropdown menu)
- Provincia:** Mendoza (dropdown menu)

At the bottom of the form are two buttons: a white button labeled "Volver" and a blue button labeled "Guardar".

Figura 159: Pantalla que nos muestra como editar el perfil desde la perspectiva del organizador.

## Editar datos



Ingrese su nombre  
**Jose**

Ingrese su teléfono o celular  
**2644638016**

Ingrese su apellido  
**aranciba**

Ingrese su DNI  
**40779581**

Buscar deporte...

Deporte seleccionado  
**Futbol-Tenis**

País  
**Argentina**

Provincia  
**San Juan**

[Volver](#)

[Guardar](#)

Figura 160: Pantalla que nos muestra como editar el perfil desde la perspectiva del árbitro.



**Editar datos**

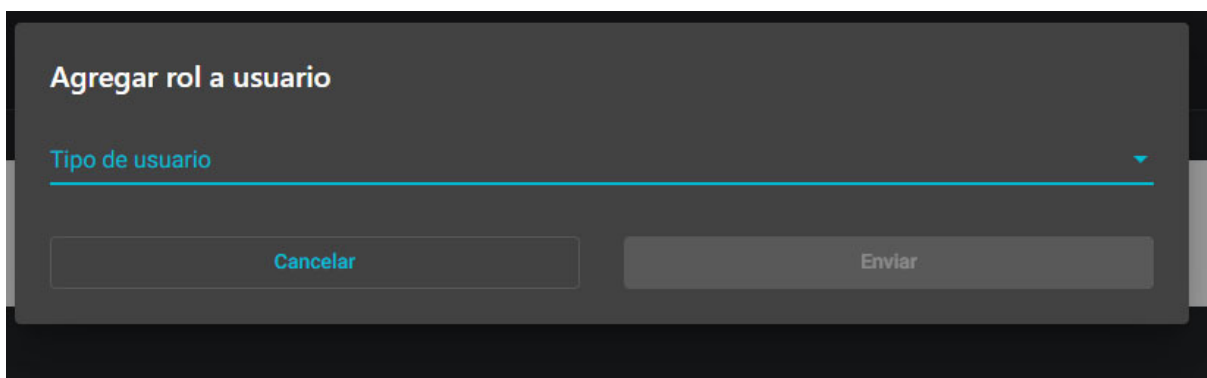
Ingrese su nombre  
**Jose**

Ingrese su teléfono o celular  
**2644638016**

Ingrese su CUIT  
**20407795815**

Volver Guardar

Figura 161: Pantalla que nos muestra como editar el perfil desde la perspectiva del patrocinador.



**Agregar rol a usuario**

Tipo de usuario

Cancelar Enviar

Figura 162: Agregar rol a usuario propio

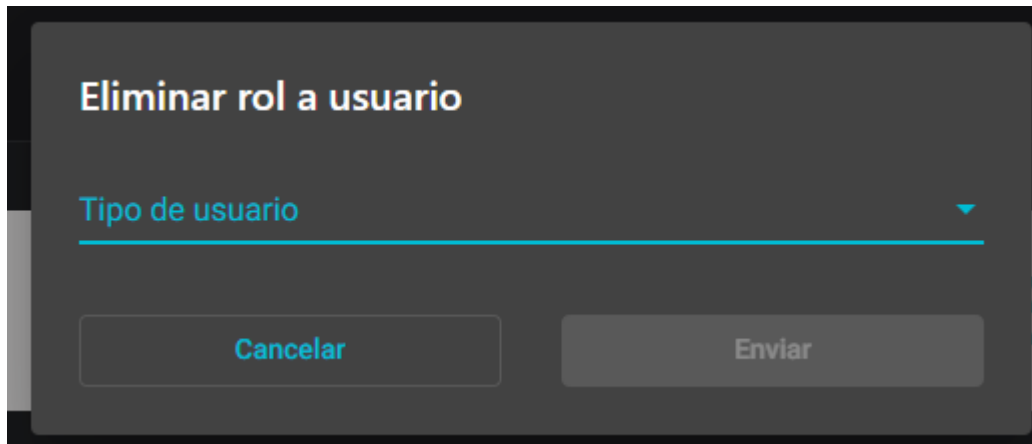


Figura 163: Quitar rol a usuario propio

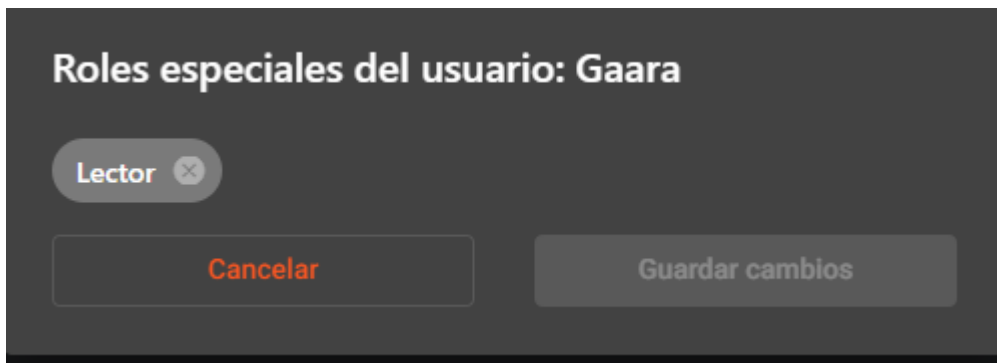


Figura 164: Quitar rol a usuario como administrador

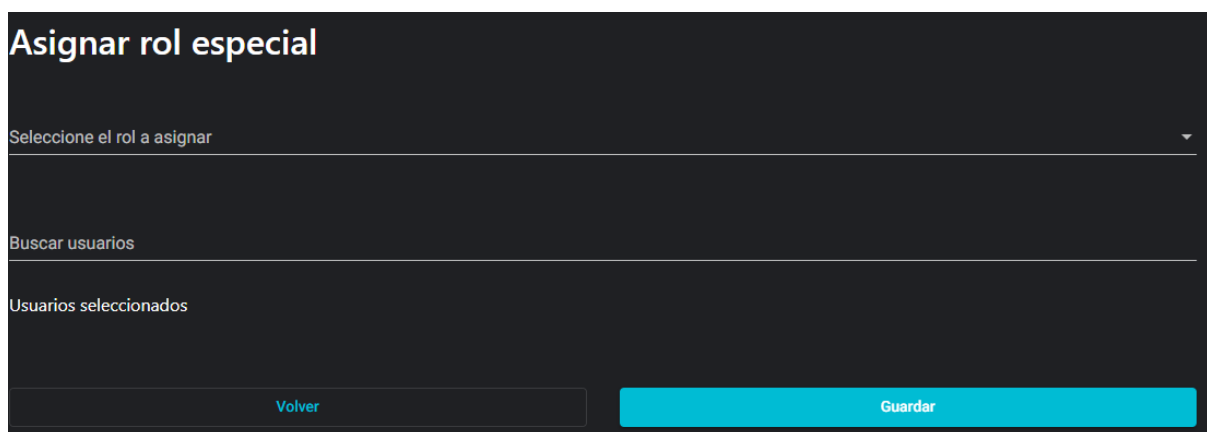


Figura 165: Agregar rol a usuario como administrador

## Editar datos



Ingrese su nombre

Jose

Ingrese su teléfono o celular

2644638016

Ingrese su domicilio

Algo

Ingrese su CUIT

12312312312

País

Argentina

Provincia

San Juan

Departamento

Santa Lucía

Localidad

ALTO DE SIERRA

## Seleccione la ubicación del club

Buscar...

Figura 166: Pantalla que nos muestra como editar el perfil desde la perspectiva del club.

Ingrese el nombre del equipo \*  
linces

Ingrese una descripción del equipo \*  
que equipo?

### Seleccionar integrantes \*

Buscar integrante  
troy bolton

No se encontró ningún registro

Integrantes seleccionados

starkloffnacho@gmail.com

Volver Guardar

Figura 167: Pantalla de alta y modificación de equipos

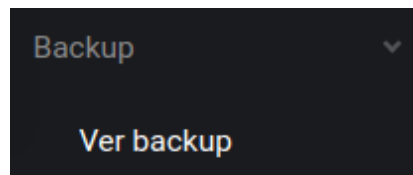


Figura 168: Apartado de backup en el sidebar del administrador

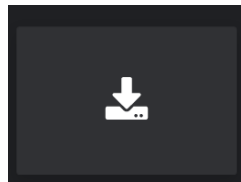


Figura 169: Botón para realizar un backup

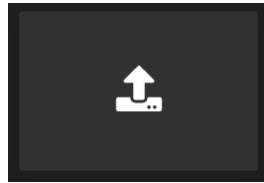


Figura 170: Botón para restaurar un backup

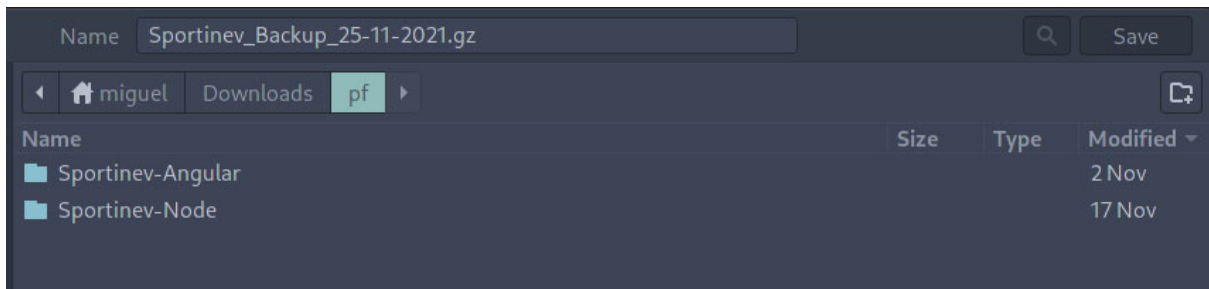


Figura 171: Imagen que nos muestra la ubicación de guardado del backup

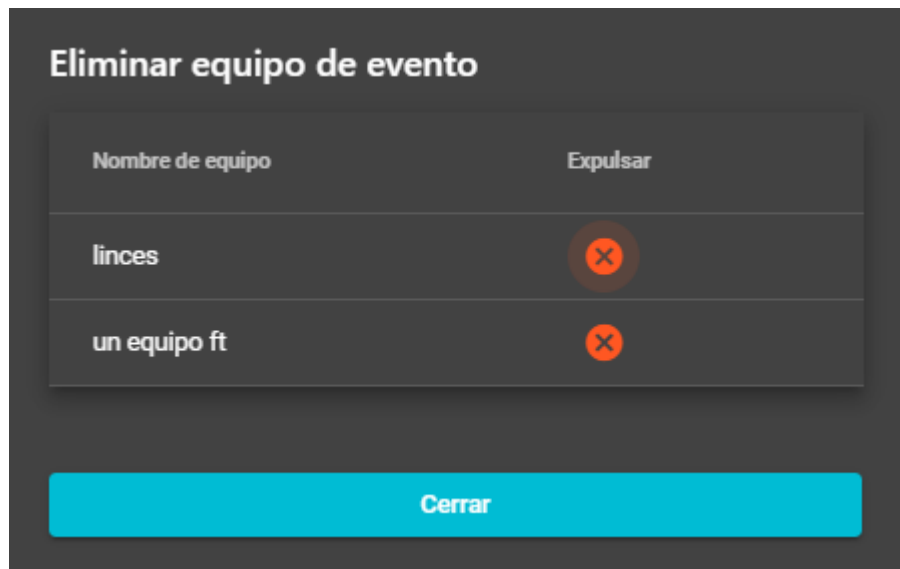


Figura 172: Ventana de eliminación de equipos del evento

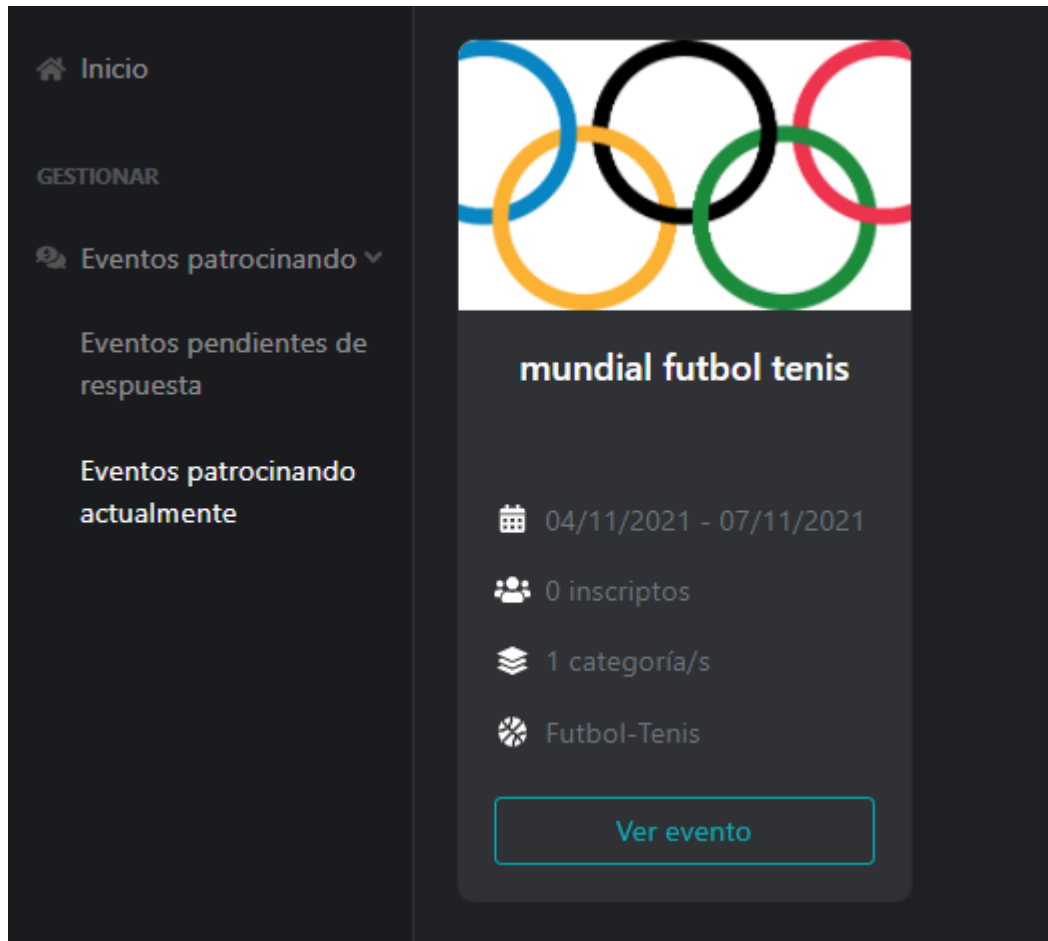


Figura 173: Visualización de eventos patrocinados actualmente

**Mejores equipos:**





Nombre	Descripción	Ver integrantes	Puntos
Boca Unidos	Un equipo unido		6
Las Estrellas	Estrelladas		3
TEST	TEST		0
STARK	TEST		0

Figura 174: Visualización de mejores equipos



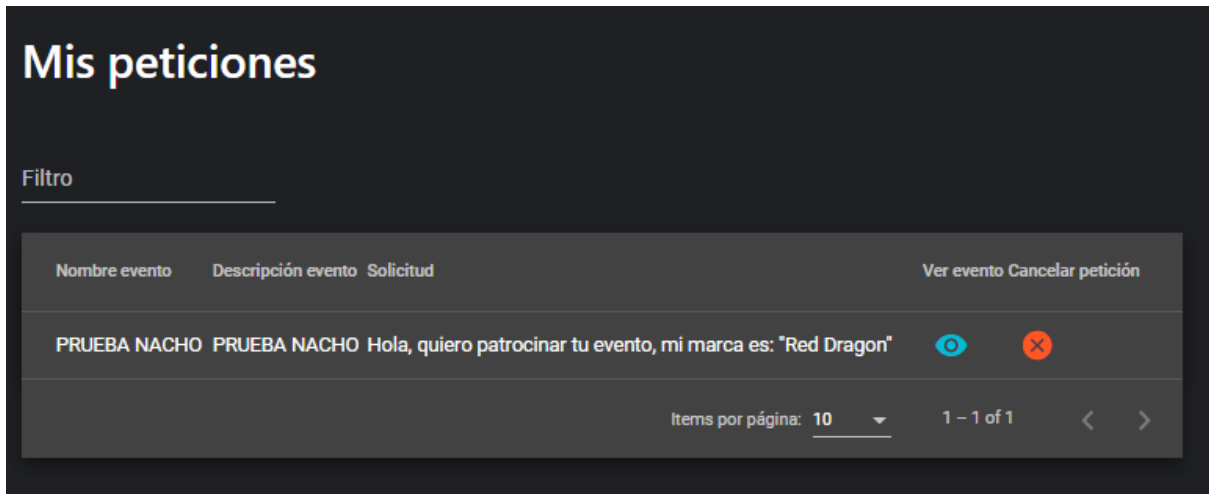


Figura 175: Visualización de eventos pendientes de respuesta de patrocinio.



Figura 176: Inscribirse a un equipo como deportista

The image shows a dark-themed mobile application interface for registering as an athlete. At the top, there is a field for 'Hora de preferencia hasta' (Preferred time until) with a clock icon and the value '12:05'. Below this is an 'Observaciones' (Observations) field containing the text 'hola'. The next section is 'Seleccionar medios de pago' (Select payment methods), with a radio button selected for 'Efectivo' (Cash). The following section is 'Seleccionar equipo a inscribir' (Select team to register), featuring a dropdown menu labeled 'Equipo' with the selected option 'lincs'. At the bottom, there are two buttons: 'Cancelar' (Cancel) and 'Enviar' (Send).

Figura 177: Inscribirse a un equipo como deportista



**Ingrese sus redes sociales**

Ingrese su link al perfil de twitter

---

Ingrese su link al perfil de Instagram

<https://www.instagram.com/clubgodoycruzofici>

---

Ingrese su link al perfil de Facebook

---

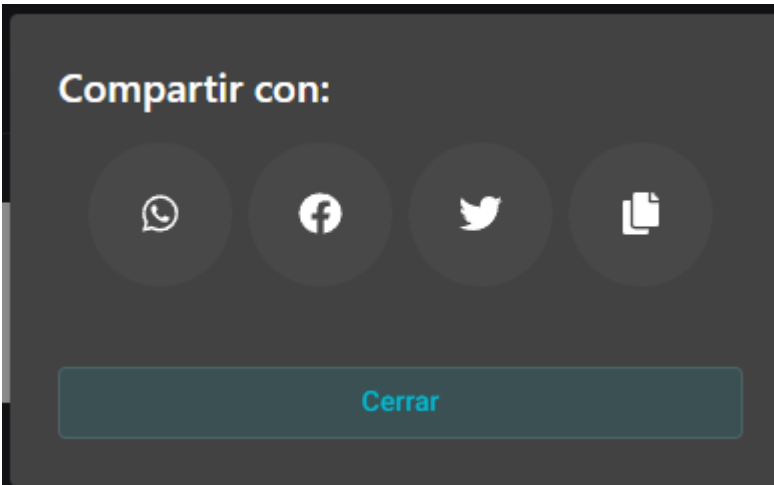
Ingrese su link al perfil de LinkedIn

<https://www.linkedin.com/in/ignacio-starkloff-07>

---

**Cancelar** **Enviar**

Figura 178: Vincular red social



**Compartir con:**

WhatsApp Facebook Twitter Documento

**Cerrar**

Figura 179: Compartir equipo con una red social

## 5. Modelo de datos

Diagramas de transición de estados

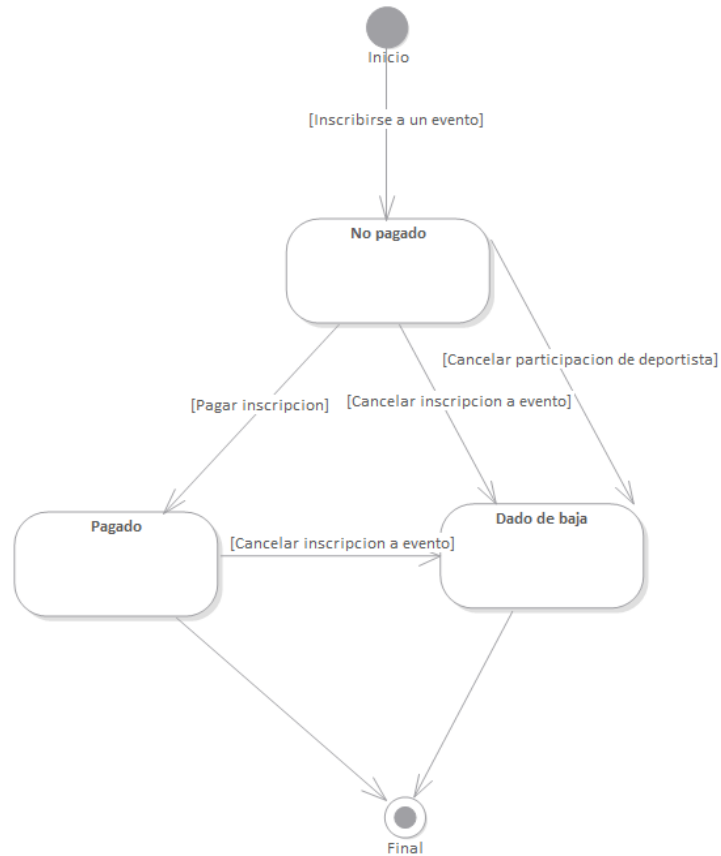


Figura 180: Diagrama de transición de estados de la inscripción

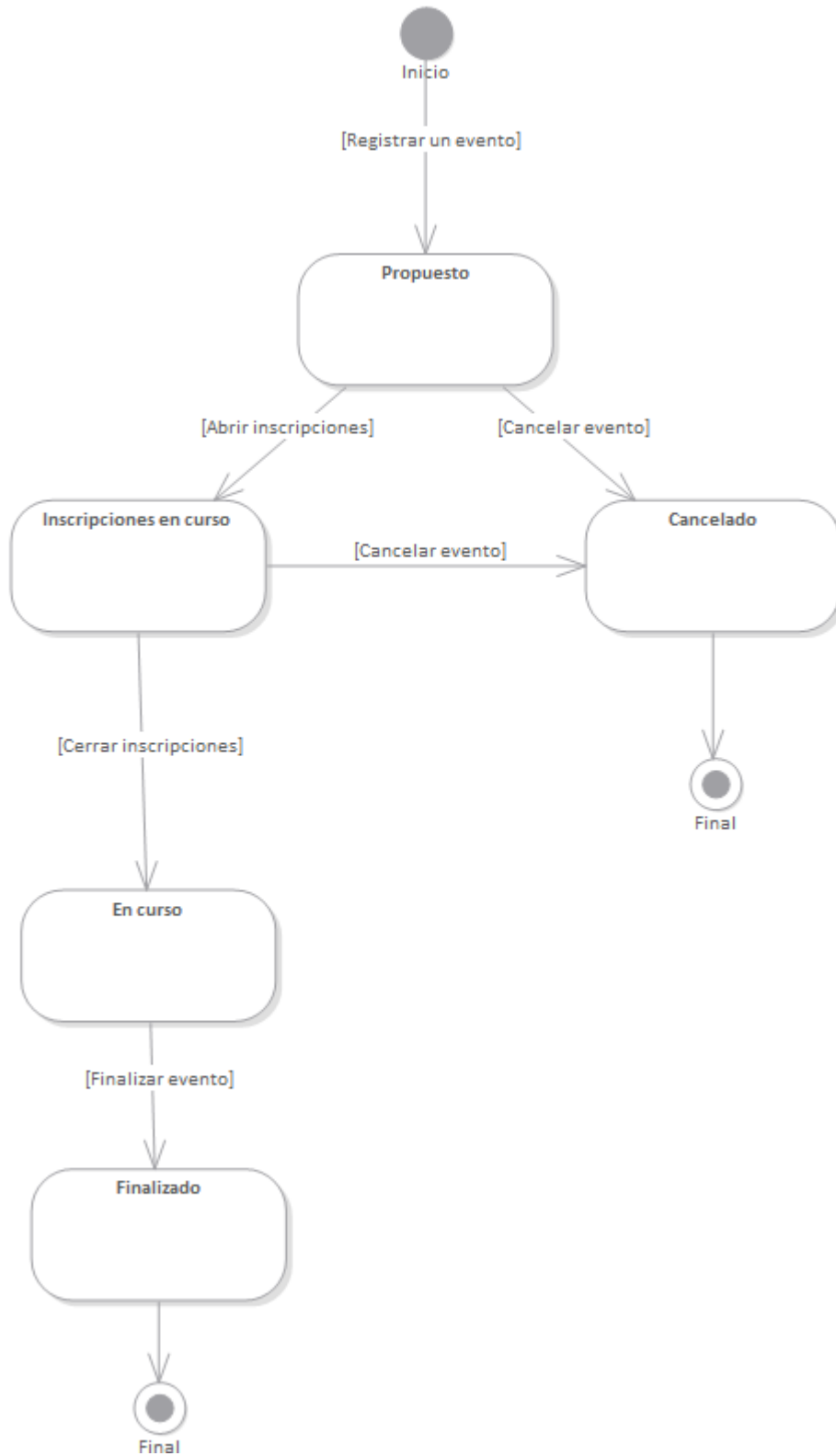


Figura 181: Diagrama de transición de estados del evento.

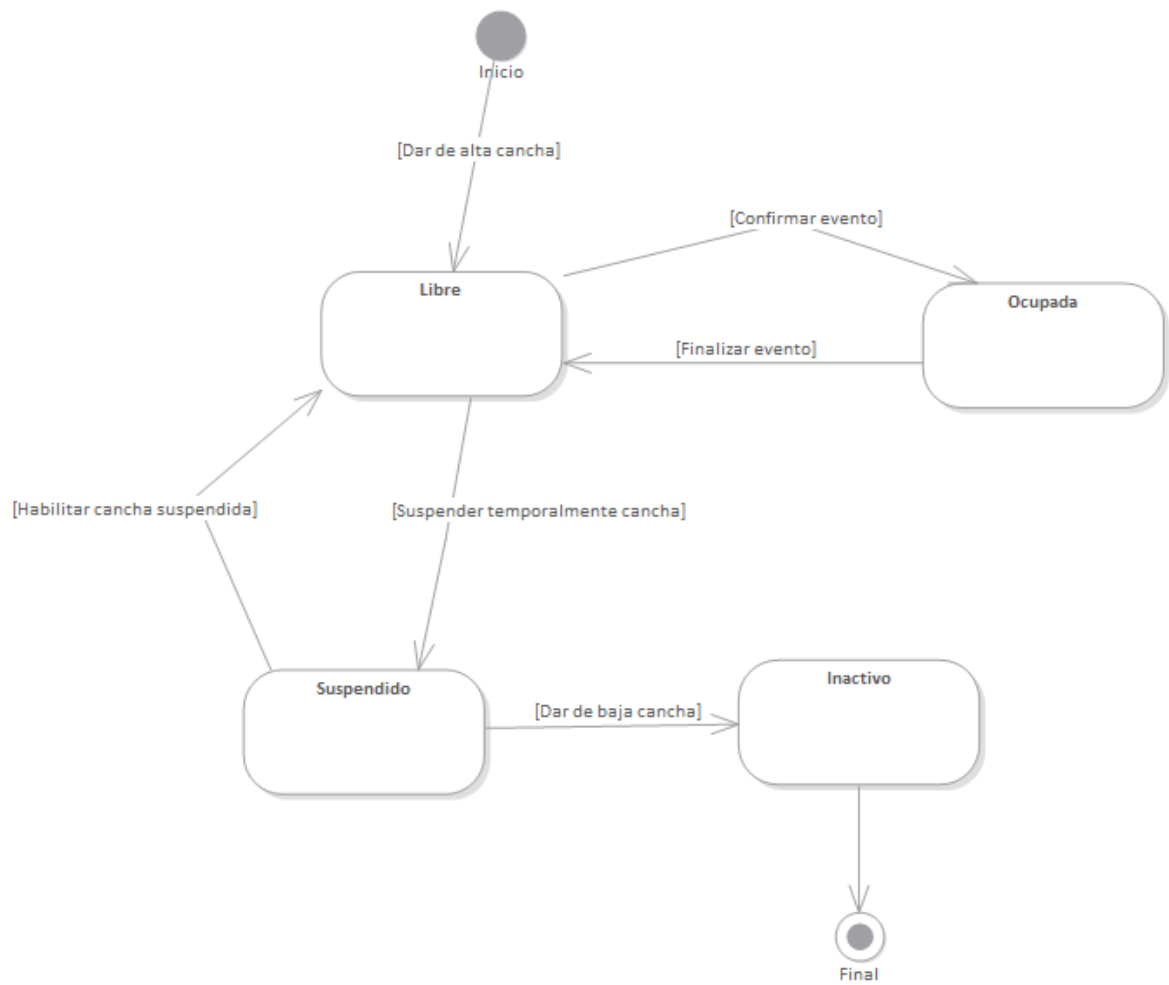


Figura 182: Diagrama de transición de estados de las canchas

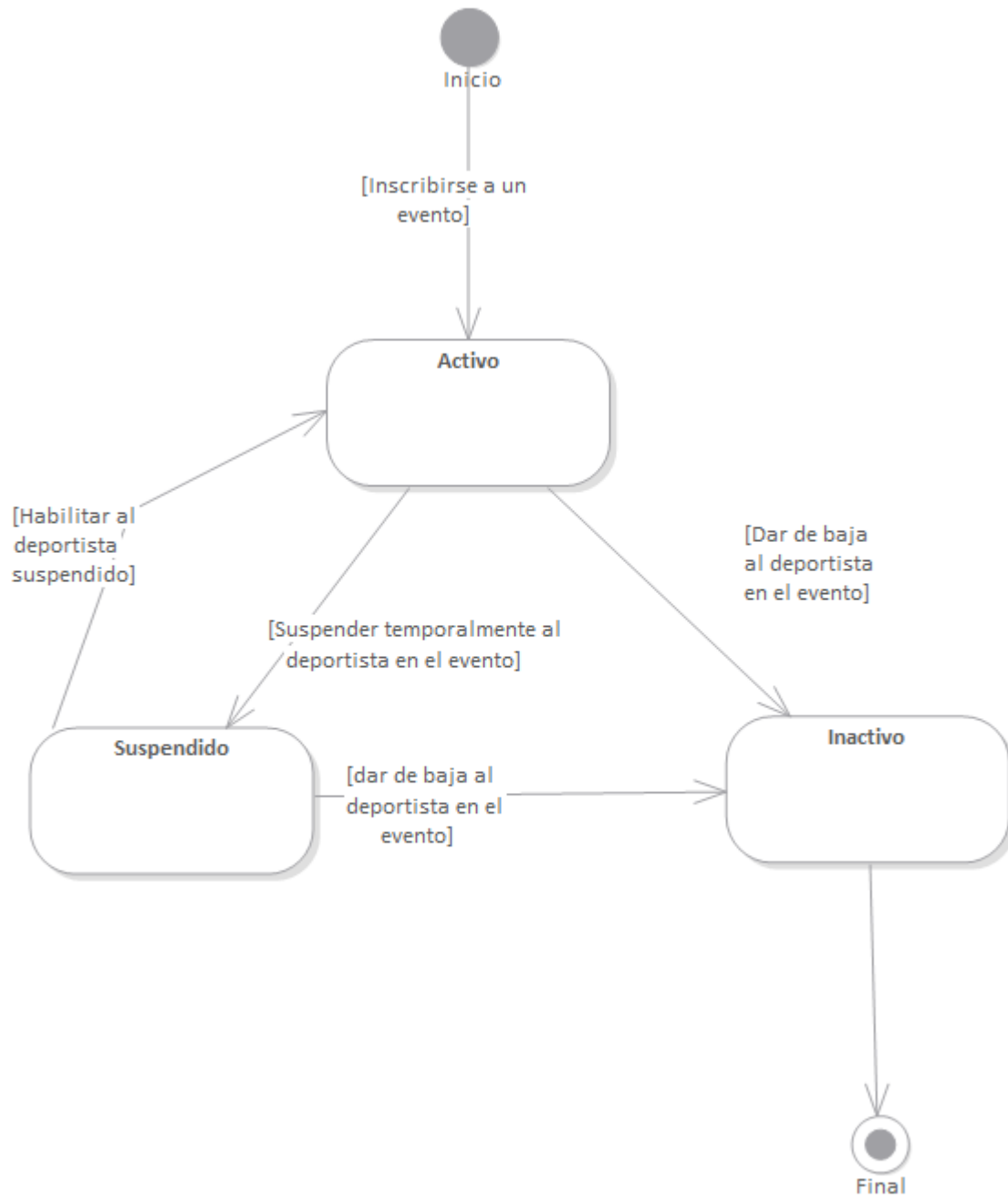


Figura 183: Diagrama de transición de estados de los deportistas en el evento

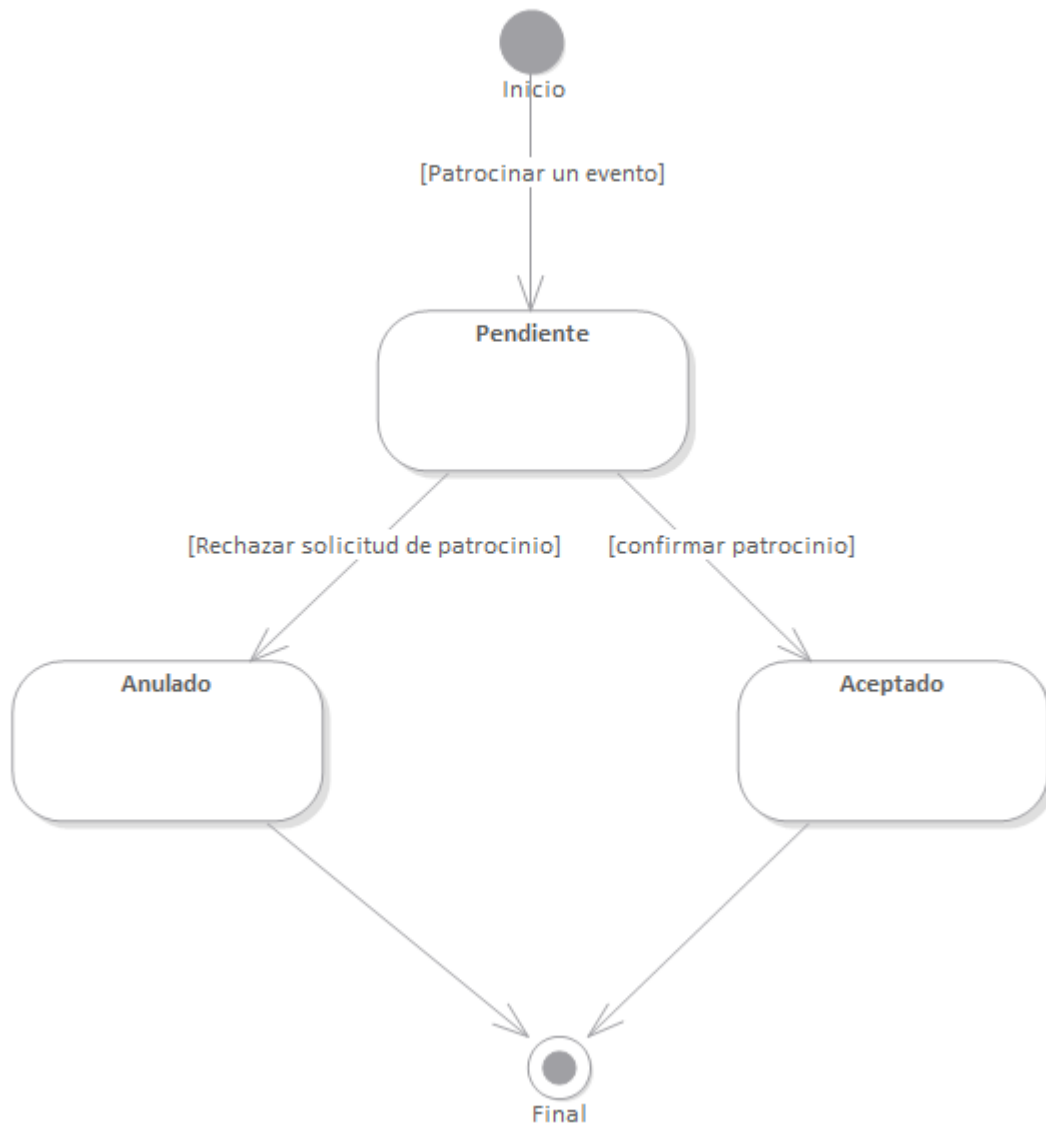


Figura 184: Diagrama de transición de estados de los patrocinadores en el evento

ANEXO 2: Diagrama de casos de uso

ANEXO 3: Diagrama de clases





# Trabajo Práctico 2

## Planificación de Proyectos Informáticos

# CAPÍTULO I: Actividades

## 1.1 Definición y descripción de actividades.

### 1.1.0 Definición del sistema

Entregable: Documentación "TP 1" en carpeta Proyecto Final / TP 1 en Google Drive.

Fecha: 04/05/2021

Sportinev es un sistema de gestión de eventos deportivos cuyo objetivo es desarrollar un sistema de información que permita a los organizadores crear eventos y llevar un registro de ellos, conectándose con los deportistas y patrocinadores interesados. El impacto consiste en generar una red de organizaciones que posibiliten a los deportistas del sistema tener una gran variedad de opciones, generando así mayor publicidad para los eventos disputados, esto quiere decir que las organizaciones podrán exponer su información hacia los clientes para que estos puedan verse interesados.

### 1.1.1 Relevamiento General

Entregable: Documentación "TP 1" en carpeta Proyecto Final / TP 1 en Google Drive.

Fecha: 29/03/2021

En un principio se decide realizar un estudio sobre como gestionar eventos deportivos, en este caso se investigan organizaciones del rubro, analizando los objetivos de estas organizaciones y los aportes que realizan sus sistemas a la sociedad o usuarios interesados en ellos.

#### Relevamiento de las organizaciones

Luego de seleccionar las organizaciones/sistemas, se inicia una investigación y estudio de todas las documentaciones que las organizaciones tienen expuestas de manera pública y luego se detecta cuáles objetivos, funciones ofrecen a sus usuarios y que tecnologías conocidas utilizan para proveer mencionadas funcionalidades.

#### Funciones detectadas

Una vez ejecutado el análisis de la documentación de cada plataforma, se detallan formalmente y técnicamente las funcionalidades que proveen las mismas, especificando cómo funciona cada módulo detectado.

Además, se efectúa una investigación de experiencia de usuario en las plataformas, para determinar de qué manera el usuario interactúa con la misma.

Entregable: Documentación "TP 1" en carpeta Proyecto Final / TP 1 en Google Drive.

Fecha: 29/03/2021

### Tecnologías de Información

En esta sección se realiza un relevamiento de todas las tecnologías que utilizan las plataformas (de las que sea posible saber) y de manera ideal conocer información acerca de sus bases de datos, arquitecturas y lenguajes de programación.

#### 1.1.2 Relevamiento detallado y análisis del Sistema.

Entregable: Documentación "TP 1" en carpeta Proyecto Final / TP 1 en Google Drive.

Fecha: 29/03/2021

Al finalizar el relevamiento general y tener conocimientos más profundos de los diferentes sistemas/organizaciones del rubro se prosigue con un análisis detallado sobre la información recopilada anteriormente (en la etapa de relevamiento).

Detalle, explicación y documentación detallada de todas las funciones seleccionadas

Se analiza y se filtra toda la información recopilada en el relevamiento general, detallando detenidamente todas las funciones ofrecidas por los sistemas relevados y así poder colocar en evidencia todas las fortalezas y debilidades de cada sistema. Información valiosa que servirá directamente en el desarrollo de nuestra plataforma, logrando de esta manera funcionalidades más originales y cubriendo necesidades no cubiertas por los otros sistemas.

#### Modelo lógico del sistema

Se realizan los modelos lógicos de cada plataforma/sistemas, de esta manera podemos representar de forma gráfica y con un lenguaje técnico todos los alcances y funcionalidades de los otros sistemas. De este modo podemos compararlos con nuestro sistema, obtener todas las diferencias y evaluar en contraposición las cualidades y defectos de cada modelo lógico.

Entregable: Documentación "TP 1" en carpeta Proyecto Final / TP 1 en Google Drive.

Fecha: 29/03/2021

Problemas y necesidades detectados en las funciones relevadas en detalle y en su entorno organizacional

Luego del análisis detallado de cada sistema relevado, se evidencian las funcionalidades que no desarrollaron y las posibles necesidades no cubiertas, para luego poder ofrecer una solución posible y tangible para dichas faltas detectadas.

#### Objetivos y alcances preliminares del nuevo Sistema

Una vez llegados a esta fase del análisis, se ponen bajo estudio las funcionalidades más importantes del proyecto, para luego elegir cuáles son los alcances del sistema, los objetivos propuestos. Así podremos ver cuáles serán las funcionalidades candidatas que pueden satisfacer las necesidades que los otros sistemas de eventos deportivos no están

logrando satisfacer.

### 1.1.3 Planificación

Entregable: Diagrama de Tiempos, Documentación "TP 2" en carpeta Proyecto Final

Fecha: 16/04/2021

#### Estimación de actividades

En esta actividad se definen las estimaciones de tiempos y de recursos asociados a cada actividad desarrollada a lo largo del proyecto

#### Selección de metodologías

Se investiga y se selecciona la metodología (o marco de trabajo) que es más conveniente y aplicable al proyecto a lo largo del desarrollo, además teniendo en cuenta que la metodología debe ser cómoda para el equipo de trabajo.

#### Definición del equipo de trabajo

En esta etapa se elige la estructura del equipo de trabajo, asignando a cada integrante del equipo uno o más roles dentro del equipo, para cumplir con todas las necesidades de desarrollo que demanda el proyecto y el proceso de diseño - implementación.

#### Definición de tecnologías

Luego de elegir los roles y las metodologías para el proceso de desarrollo del proyecto, se eligen las tecnologías para el proyecto, eligiendo cuáles tecnologías se adaptan mejor a las necesidades del mismo y a la metodología de trabajo adoptada. Se elegirán los lenguajes de programación, gestores de bases de datos, Freshworks, arquitecturas, plataforma de despliegues, circuitos de integración continua, etc.

#### Definición de herramientas de comunicación

Se elegirán las plataformas y la manera más viable de comunicación para el equipo, siendo las mejores herramientas las que mejor se adapten a los horarios dispares que posee un equipo de trabajo.

#### Definición de herramientas para el seguimiento de tareas

Se elige entre las opciones disponibles, alguna herramienta que permita controlar y tener un fácil seguimiento (lo más gráfico y amigable posible) de las tareas y los estados en que se encuentran las tareas de cada integrante del equipo.

#### Definición de herramientas para la gestión de configuración del software

Se eligen las herramientas correctas que integradas entre sí permiten el óptimo seguimiento de las versiones del código producto del desarrollo de los integrantes.

#### Definición y descripción de recursos para cada una de las actividades

Se analizan y se asignan los diferentes recursos a cada actividad con la finalidad de tener la mejor asignación posible según estimación y esfuerzo que conlleva cada tarea.

#### Realizar estudio de factibilidad.

Realizar diversos estudios que demuestran según distintos criterios si un proyecto es factible, se evalúan los siguientes aspectos:

- Factibilidad Económica
- Factibilidad Legal
- Factibilidad Técnica
- Factibilidad Operativa

#### Costos de infraestructura y tecnologías

Se planifican los costos asociados con las tecnologías que se utilizan en el proyecto como licencias de desarrollo, licencias de programas terceros utilizados por el equipo, costos de plataformas que se utilicen para hacer el despliegue de las aplicaciones.

#### Análisis de riesgos

Se efectúa un análisis de las fortalezas, oportunidades, debilidades y amenazas del sistema. Además, se ejecuta un análisis de los riesgos que comprende el sistema, revisando las amenazas y el impacto que comprenden en el sistema.

#### Análisis de impacto ambiental

Se ejecuta un análisis de cómo el sistema puede impactar en el ambiente de los deportistas una vez que los mismos sean usuarios del sistema y en las ventajas/desventajas que tiene el sistema a la hora de efectuar inscripciones a eventos deportivos.

### 1.1.4 Diseño

Entregables: Diagrama de Clases, Diagrama de Casos de Uso, Documentación "TP 1" en carpeta Proyecto Final.

Fecha: 10/05/2021

#### Objetivos y alcances definitivos del nuevo Sistema.

Se explican de manera detallada y exhaustiva los objetivos y los alcances definitivos, además se refina y se filtra toda la información de las funcionalidades pensadas para el sistema.

### Salidas del sistema

Se definen las salidas de los servicios que tendrá el sistema, o explicado de diferente manera, en esta actividad se describirán los datos que llegan a la capa de aplicación que será encargada de mostrar los datos.

### Mockups

En esta actividad se diseñan las interfaces con las que interactúa el usuario, cada caso de uso tendrá su propia interfaz para el mejor entendimiento de cómo el usuario interactúa con el sistema.

### Modelo de datos

En esta actividad se hace una representación mediante diferentes diagramas sobre como se van a modelar los datos del sistema (objetos, tablas)

### Definición de casos de uso

Se determinan los casos de uso del sistema, estos mismos estarán divididos por módulos y según los requerimientos.

### Definición de historias de usuario

Se definen las historias de usuario, donde se define lo que el usuario solicita, por qué realiza esa solicitud y sus criterios de aceptación para que la historia se dé por terminada.

### Estimación de historias de usuario

Se estima la duración de cada historia de usuario (lo más exacta y aproximada posible) y se priorizan para luego seguir un orden por importancia en su desarrollo.

### Definición de Ceremonias de la metodología ágil

La realización del proyecto será guiada por Sprints. Se define una cantidad estimativa de Sprints que tendrá el proyecto y que tareas, actividades y procesos efectuará cada integrante de grupo en estos Sprints. Los objetivos y los alcances de los Sprints se irán refinando a medida que el equipo vaya iniciando en los mismos.

Los Sprints definen una cantidad de tiempo en el que el equipo deberá deliberar un entregable.

Todos los sprints inician con una ceremonia donde se planea y se prioriza con detalle y precisión lo que se hará en dicho Sprint, esta ceremonia se denomina Sprint Planning.

Además, se ejecutarán reuniones diarias, donde los integrantes del equipo de desarrollo deberán comentar en lo que han estado trabajando el día anterior a la reunión. Comentarán lo que realizarán en el transcurso del día y si tienen algún impedimento que les está bloqueando el avance de alguna tarea, con el fin de que el equipo pueda ayudarlo a desbloquearse.

Cuando se finaliza un sprint, el equipo organizará una reunión llamada Retrospectiva del Sprint, donde se analizarán las cosas buenas y malas del sprint finalizado y se tomará un tiempo para organizar mejor las futuras tareas del sprint siguiente.

Los sprints tienen 7 días y se realizarán 8 sprints para desarrollar los módulos del sistema:

#### Diseño de módulo seguridad

Diseñar los diagramas de caso de uso con sus escenarios correspondientes, con las funcionalidades relacionadas con el módulo de seguridad, como registro, login, manejo de permisos, manejo de tokens de usuarios, restricción de acciones en el servidor, confirmación de cuenta y cambio de contraseña.

Diseño de las clases necesarias para lograr las funcionalidades del módulo de seguridad.

Diseño de escenarios y casos de pruebas para el módulo de seguridad.

#### Diseño de módulo eventos

Diseñar los diagramas de caso de uso con sus escenarios, planteando como crear eventos, modificarlos, dar de baja uno, relacionar categorías para los eventos, funcionalidades relacionadas con la creación y manejo de partidos, también la función de mostrar los eventos en la plataforma para cualquier persona.

Diseño de las clases necesarias para lograr las funcionalidades del módulo de eventos.

Diseño de escenarios y casos de pruebas para el módulo de eventos.

#### Diseño de módulo reportes

Diseñar los diagramas de caso de uso con sus escenarios correspondientes, con las funcionalidades relacionadas con el módulo de reportes, como los diferentes reportes que los distintos usuarios pueden ver en la plataforma.

Diseño de las clases necesarias para lograr las funcionalidades del módulo de reportes.

Diseño de escenarios y casos de pruebas para el módulo de reportes.

#### Diseño de módulo publicidad

Diseñar los diagramas de caso de uso con sus escenarios correspondientes, con las funcionalidades relacionadas con el módulo de publicidad, como generar las publicidades para los distintos interesados, la posición de las publicidades, las imágenes elegidas.

Diseño de las clases necesarias para lograr las funcionalidades del módulo de publicidad.

Diseño de escenarios y casos de pruebas para el módulo de publicidad.

#### Diseño de módulo inscripción

Diseñar los diagramas de caso de uso con sus escenarios correspondientes, con las funcionalidades relacionadas con el módulo de inscripción, como solicitar jugar un evento en un club determinado, posibilidad de que los equipos se inscriban al evento, posibilidad de invitar árbitros a los eventos y además posibilidad de solicitar patrocinios a los eventos.

Diseño de las clases necesarias para lograr las funcionalidades del módulo de inscripción.

Diseño de escenarios y casos de pruebas para el módulo de inscripción.

#### Diseño de módulo pagos

Diseñar los diagramas de caso de uso con sus escenarios correspondientes, con las funcionalidades relacionadas con el módulo de pagos, como posibilidad de registrar pagos en efectivo o por mercado pago.

Diseño de las clases necesarias para lograr las funcionalidades del módulo de pagos.

Diseño de escenarios y casos de pruebas para el módulo de pagos.

#### Diseño de módulo feedback

Diseñar los diagramas de caso de uso con sus escenarios correspondientes, con las funcionalidades relacionadas con el módulo de feedback, como posibilidad de realizar un feedback a un evento al cual se haya participado previamente.

Diseño de las clases necesarias para lograr las funcionalidades del módulo de feedback.

Diseño de escenarios y casos de pruebas para el módulo de feedback.

#### Diseño de módulo parametrización

Diseñar los diagramas de caso de uso con sus escenarios correspondientes, con las funcionalidades relacionadas con el módulo de parametrización, como la posibilidad de cargar países, provincias, localidades, deportes, tipo de organizadores, roles especiales y demás datos configurables en el sistema.

Diseño de las clases necesarias para lograr las funcionalidades del módulo de parametrización.

Diseño de escenarios y casos de pruebas para el módulo de parametrización.

#### Diseño de módulo perfil

Diseñar los diagramas de caso de uso con sus escenarios correspondientes, con las funcionalidades relacionadas con el módulo de perfil, como la visualización de los perfiles, la modificación de los datos y las funcionalidades relacionadas con los perfiles de los usuarios.

Diseño de las clases necesarias para lograr las funcionalidades del módulo de perfil.

Diseño de escenarios y casos de pruebas para el módulo de perfil.

#### Diseño de pruebas de integración

Se diseñan las pruebas que aseguran la correcta integración de los distintos módulos del sistema y componentes, los cuales deberán trabajar de la manera correcta para poder superar las pruebas diseñadas.

#### Diseño de pruebas de seguridad

Se diseñan pruebas que mediante escaneos de vulnerabilidades y herramientas para auditar la seguridad de los sistemas informáticos probarán la seguridad del sistema, pruebas que se realizarán con respecto a redes, aplicación e infraestructura.



## Diseño de pruebas de stress/carga

A través de herramientas que prueban la carga de los sistemas simulando peticiones por segundo que harían miles de usuarios, se prueba la cantidad de carga máxima que soporta. El sistema antes de empezar a fallar en la devolución de datos o cuando la demora sea excesiva y no produzca una buena experiencia al usuario.

### 1.1.5 Desarrollo e implementación

Entregables: Demo Aplicación, Manual de Usuario, Documentación "TP 1" en carpeta Proyecto Final.

Fecha: 02/11/2021

#### Desarrollo del formato sobre el cual vamos a trabajar

Se define el formato de trabajo que el equipo deberá respetar para poder llevar una estructura de desarrollo y orden aceptable, logrando así una buena calidad en el producto y una buena minimización de errores.

A medida que cada integrante del equipo vaya desarrollando las funcionalidades del sistema irá documentando dichas funcionalidades con la finalidad de tener información acerca de cada porción de funcionalidad (técnica y de negocio).

Una vez finalizado los módulos el equipo realizará las pruebas unitarias (estas pruebas se irán desarrollando a medida que se desarrollen las funcionalidades), pruebas de integración, de seguridad y de carga para verificar que la calidad del sistema es la esperada.

#### Programación, pruebas y documentación de los módulos

Se lleva a cabo la codificación de los módulos probándolos íntegramente cada desarrollador.

Una vez finalizado los módulos, el equipo ejecutará las pruebas unitarias o exploratorias si es necesario, pruebas de integración, de seguridad y de carga para comprobar que la calidad del sistema es la esperada.

#### Realizar el manual de usuario del Sistema completo

Se crea un manual que facilita al usuario la interacción con el sistema, donde pueda ir a consultar qué y cómo se llevan a cabo las operaciones en el sistema.

#### Implementación del Sistema.

Una vez finalizado el desarrollo del sistema se procede a ingresarlo al mercado, a través de diferentes técnicas de marketing, brindando tutoriales y soporte técnico a los usuarios.

#### Realizar capacitación a usuarios

En esta actividad se proporcionará a los usuarios los conocimientos necesarios para poder utilizar correctamente y de la mejor manera *Sportinev*, desarrollando un plan que los acompañe y les permita entender de la mejor manera el uso de cada funcionalidad.

En este plan se especifican los usuarios, objetivos, formas y técnicas de capacitación, actividades, recursos y documentación que necesite el usuario para realizar la capacitación.

#### Demo del sistema y ensayo de exposición

La presentación fue llevada a cabo el día 9 de noviembre del 2021, en la misma se mostró la ejecución e integración de los distintos módulos del sistema desarrollado, además se nos impartió las pautas a seguir para la exposición del día 16 de noviembre.

#### Exposición anual de Proyecto de Sistemas

En la exposición efectuada el día 16 de noviembre del 2021, se llevó a cabo un vivo por YouTube con el fin de mostrar a familiares, amigos y demás personas interesadas en el sistema desarrollado durante el año lectivo. Para cada uno de los grupos se asignaron 20 minutos, en los cuales se mostró un video explicativo del sistema sportinev, luego se efectuó una ronda de preguntas de una duración de 4 minutos donde se resolvían las dudas de los espectadores.

## 1.2 Diagrama de tiempos

### ANEXO 1: Diagrama de tiempo de Sportinev Gantt

## CAPÍTULO II: Organización para la ejecución del proyecto.

### 2.1 Equipo de trabajo

#### 2.1.1 Metodología a adoptar por el equipo de trabajo

La **metodología Scrum** es un marco de trabajo o framework que se utiliza dentro de equipos que manejan proyectos complejos. Es decir, se trata de una **metodología de trabajo ágil** que tiene como finalidad la entrega de valor en períodos cortos de tiempo y para ello se basa en tres pilares: la transparencia, inspección y adaptación.

Se empleará esta metodología por su gran flexibilidad al cambio de requerimientos que pueda haber durante el desarrollo del proyecto, se adapta muy bien a equipos de números reducidos y es eficaz en cuanto a la tangibilidad del producto que se desarrolla. Habiendo entregas periódicas de incrementos de funcionalidades y ceremonias que hacen el trabajo en equipo mucho mejor por la comunicación que sustenta usar esta metodología.

#### 2.1.2 Tecnologías a usar por el equipo de trabajo

A continuación se describen las tecnologías utilizadas en el proyecto:

Angular



Figura 185: Logo de Angular

Angular es un framework open source desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página, las webs SPA (Single Page Application).

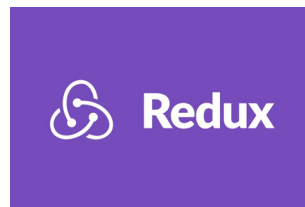
Angular separa completamente el frontend y el backend en la aplicación, evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón MVC (Modelo-Vista-Controlador) asegurando los desarrollos con rapidez, a la vez que posibilita modificaciones y actualizaciones.

En una web SPA aunque la velocidad de carga puede resultar un poco lenta la primera vez que se abre, navegar después es totalmente instantáneo, ya que se ha cargado toda la página de golpe.

Solamente es una ruta la que se tiene que enviar el servidor, y Angular lo que hace 'por debajo' es cambiar la vista al navegar para que dé la apariencia de una web normal, pero de forma más dinámica.

Entre otras ventajas, este framework es modular y escalable adaptándose a nuestras necesidades y al estar basado en el estándar de componentes web, y con un conjunto de interfaz de programación de aplicaciones (API) permite crear nuevas etiquetas HTML personalizadas que pueden reutilizarse.

## Redux



*Figura 186:* Logo de Redux

Redux es una librería JavaScript que emite actualizaciones de estado en respuesta a acciones, con la peculiaridad de realizar dichas modificaciones a través de objetos sencillos, que reciben el nombre de acciones, y no a través de cambios directos en el estado.

Redux es un contenedor predecible del estado de aplicaciones JavaScript.

Te ayuda a escribir aplicaciones que se comportan de manera consistente, corren en distintos ambientes (cliente, servidor y nativo), y son fáciles de probar. Además de eso, provee una gran experiencia de desarrollo, gracias a edición en vivo combinado con un depurador sobre una línea de tiempo.

Puedes usar Redux combinado con React, o cualquier otra librería de vistas. Es muy pequeño y no tiene dependencias.

## Bootstrap



*Figura 187:* Logo Bootstrap

Bootstrap es un framework que combina CSS y JavaScript para estilizar los elementos de una página HTML. Permite mucho más que, simplemente, cambiar el color de los botones y los enlaces.

Esta es una herramienta que proporciona interactividad en la página, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso y más.

Además de todas las características que ofrece el framework, su principal objetivo es permitir la construcción de sitios web responsive para dispositivos móviles. Esto significa que las páginas están diseñadas para funcionar en desktop, tablets y smartphones, de una manera muy simple y organizada.

El stack tecnológico de backend está compuesto por:

Node.js



Figura 188: Logo de nodeJS

En la parte de los servicios del sistema utilizaremos Node.js®. Es una tecnología ideada como un entorno de ejecución de JavaScript orientado a eventos asíncronos, diseñada fundamentalmente para crear aplicaciones network escalables.

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue producido con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, servidores web.

Express.js



Figura 189: Logo de express

Express.js es un framework para Node.js, es software gratuito y de código abierto bajo la licencia MIT. Está diseñado para crear aplicaciones web y API. Se le ha llamado el marco de servidor estándar de facto para Node.js.

Es un framework inspirado en Sinatra, lo que significa que es relativamente mínimo con muchas características disponibles como complementos. Express es el componente de backend utilizado en metodologías como MEAN, MERN o MEVN, junto con el software de base de datos MongoDB y un marco o biblioteca de frontend de JavaScript.

## Kubernetes



Figura 190: Logo de Kubernetes

Kubernetes es una plataforma open source para administrar cargas de trabajo y servicios, facilitando la automatización y eliminando muchos de los procesos manuales durante la implementación y escalabilidad de las aplicaciones en contenedores. En este sentido, nos ofrece una mayor simpleza en plataformas como servicio (PaaS) y una mayor flexibilidad en la infraestructura como servicio (IaaS), permitiendo la portabilidad entre proveedores de infraestructura. Se puede desplegar en diversos entornos cloud y soporta múltiples runtimes de contenedores (docker, rkt, cri-o o frakti).

Aunque en el origen de esta interesante plataforma portable y extensible fue Google el encargado del desarrollo y diseño, en 2014 el gigante tecnológico decidió liberar el proyecto y donarlo a la Cloud Native Computing Foundation (Linux Foundation). Al igual que Docker, Kubernetes se ha convertido en uno de los estándares como gestor de contenedores de software dentro de la industria.

## Docker

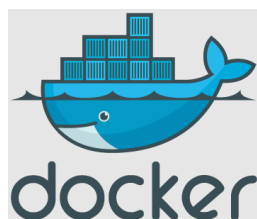


Figura 191: Logo de Docker

Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.

Docker utiliza características de aislamiento de recursos del kernel Linux, tales como cgroups y espacios de nombres (namespaces) para permitir que "contenedores" independientes se ejecuten dentro de una sola instancia de Linux, evitando la sobrecarga de iniciar y mantener máquinas virtuales.

El soporte del kernel Linux para los espacios de nombres, aísla la vista que tiene una aplicación de su entorno operativo, incluyendo árboles de proceso, red, ID de usuario y sistemas de archivos montados, mientras que los cgroups del kernel proporcionan aislamiento de recursos, incluyendo la CPU, la memoria, el bloque de E/S y de la red. Desde la versión 0.9, Docker incluye la biblioteca libcontainer como su propia manera de utilizar directamente las facilidades de virtualización que ofrece el kernel Linux, además de utilizar las interfaces abstraídas de virtualización mediante libvirt, LXC (Linux Containers) y systemd-nspawn.

## MongoDB



Figura 192: Logo de mongoDB

En lo que respecta a la base de datos usaremos MongoDB, el cual es un motor de base de datos no relacional que posee muchas ventajas porque al ser un esquema documental, la información que se almacena no se guarda bajo un formato rígido de almacenamiento.

En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

MongoDB es una base de datos adecuada para su uso en producción y con múltiples funcionalidades.

Esta base de datos se emplea mucho en la industria contando con implantaciones en empresas como MTV Network, Craigslist, Foursquare, etc.

El código fuente está disponible para los sistemas operativos Windows, GNU/Linux, OS X y Solaris.

## Redis



*Figura 193* Logo de redis

Redis, que significa Remote Dictionary Server (servidor de diccionarios remoto), es un rápido almacén de datos clave-valor en memoria de código abierto que se puede utilizar como base de datos, caché, agente de mensajes y cola. Redis ofrece ahora tiempos de respuesta inferiores al milisegundo, lo que permite que se realicen millones de solicitudes por segundo para aplicaciones en tiempo real de videojuegos, tecnología publicitaria, servicios financieros, sanidad y IoT.

Redis es una opción muy habitual en aplicaciones de almacenamiento en caché, administración de sesiones, videojuegos, tablas de clasificación, análisis en tiempo real, datos geo espaciales, servicios de vehículos compartidos, chat/mensajería, streaming de contenido multimedia y publicación/suscripción.

## Mongoose



elegant **mongodb** object modeling for **node.js**

*Figura 194:* Logo de mongoose

Mongoose es una herramienta de modelado de objetos de MongoDB diseñada para funcionar en un entorno asíncrono. Mongoose admite promesas y callbacks.

## Cloudinary



*Figura 195:* Logo de Cloudinary



Cloudinary es una empresa de tecnología SaaS con sede en Santa Clara, California, con oficinas en Israel, Inglaterra, Polonia y Singapur. La empresa ofrece servicios de gestión de imágenes y vídeos basados en la nube

VS code



*Figura 196:* Logo de vs code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

WebStorm



*Figura 197:* Logo de WebStorm

WebStorm es un entorno de desarrollo integrado para JavaScript y las tecnologías relacionadas. Al igual que otros IDE de JetBrains, hace que su experiencia de desarrollo sea más agradable, automatiza las tareas repetitivas y le ayuda a gestionar las tareas complejas con facilidad.

El stack tecnológico de pruebas está compuesto por:

Jmeter:



*Figura 198:* Logo de JMeter

JMeter es un proyecto de Apache que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el rendimiento de una variedad de servicios, con énfasis en aplicaciones web.

JMeter puede ser usado como una herramienta de pruebas unitarias para conexiones de bases de datos con JDBC, FTP, LDAP, Servicios web, JMS, HTTP y conexiones TCP genéricas. JMeter puede también ser configurado como un monitor, aunque es comúnmente considerado una solución ad-hoc respecto de soluciones avanzadas de monitoreo.

En nuestro proyecto empleamos jmeter para realizar las pruebas funcionales, de seguridad y principalmente como herramienta de generación diseño y ejecución de pruebas de carga y estrés, que es donde esta herramienta más se destaca.

Jasmine:



*Figura 199:* Logo de Jasmine

Jasmine es un marco de desarrollo impulsado por el comportamiento para probar código JavaScript. No depende de ningún otro marco de JavaScript. No requiere DOM. Y tiene una sintaxis clara y obvia para que pueda escribir pruebas fácilmente.

El desarrollo dividido en dos partes, front-end, la parte del sistema que interactúa directamente con el usuario, y back-end, la parte que gestiona los datos, la seguridad y la lógica de negocio.

### 2.1.3 Puestos de trabajo y cantidad de personas por puesto

- Desarrollador Frontend: 2 personas.
- Desarrollador Back-End: 3 personas.
- Ingeniero QA: 1 persona.
- DevOps: 1 persona.
- Scrum Master: 1 persona.
- DBA: 1 persona.
- Analista: 3 personas.
- Diseñador: 3 personas.

### 2.1.4 Descripción de los puestos de trabajo

Desarrollador Frontend:

**Objetivos del puesto:** Programar una interfaz que le permita al cliente ejecutar las funcionalidades del sistema de una forma rápida y sencilla.

**Funciones del puesto:**

- Desarrollar con HTML, CSS y JavaScript.
- Mantenimiento y optimización de las interfaces de usuario.
- Crear herramientas y elementos que mejoren la interacción con el sitio en cualquier dispositivo o browser.
- Gestionar el flujo de trabajo del software.
- Implementar prácticas SEO Onpage.
- Mejorar la usabilidad para el usuario – Nuevas funcionalidades y procesos.
- Convertir diseño de parte de UX/UI en realidad.
- Mejorar velocidad de carga de la web.
- Asegurar la accesibilidad de la web.

Desarrollador Back-End:

**Objetivos del puesto:** Implementar el Backend de la aplicación en función del diseño realizado.

**Funciones del puesto:**

- Experiencia práctica demostrada en el análisis, desarrollo y mantenimiento de bases de datos.
- Comprensión de los parámetros y criterios de diseño, para traducir los requerimientos en una arquitectura operativa.
- Diseño y programación de plantillas, temas e interfaces.
- Conocimientos y destrezas sobresalientes en el manejo de bases de datos (Oracle, MySQL) y tecnologías SQL y ORM (JPA, Hibernate).

- Lenguajes de scripting (Java, Javascript, Python, C++, C#).
- Experiencia en el desarrollo de aplicaciones web con frameworks relevantes (Django, Struts, Spring, etc.).
- Definir metas de rendimiento y tareas.
- Atender y solucionar incidencias de integración.

#### Ingeniero QA:

**Objetivos del puesto:** Diseñar e implementar pruebas que aumenten la calidad del sistema y reduzcan la cantidad de errores en producción.

**Funciones del puesto:**

- Crear un plan de pruebas y testing.
- Probar los programas de software de automatización.
- Simular el rendimiento del producto y evaluar los resultados.
- Identificar los problemas de los productos mediante el uso de sistemas de seguimiento de errores.
- Generar bases de datos de defectos de productos conocidos y analizar estos problemas.
- Revisar las especificaciones del producto para predecir futuros errores.
- Asesoramiento sobre el diseño de productos para reducir los posibles problemas.
- Mejorar las estrategias de pruebas.
- Asegurar que el software está listo para el público objetivo.

#### DevOps:

**Objetivos del puesto:** Automatizar los procesos de testing e implementación del sistema.

**Funciones del puesto:**

- Asegurar la integración y despliegue continuo.
- Encargado de aprovisionar los recursos o infraestructura requeridos para los proyectos.
- Seleccionar un modelo de implementación apropiado para el proyecto.
- Diseñar y ordenar los protocolos de pruebas.
- Monitorización continua de los procesos de desarrollo, operaciones y test.
- Análisis de resultados obtenidos en cada una de las fases de desarrollo.
- Comunicación de problemas de desarrollo y proposición de soluciones.
- Asignar y comunicar las distintas tareas a realizar.

#### Analista:

**Objetivos del puesto:** Efectuar el relevamiento y la definición de objetivos preliminares del sistema, así como la captura de requerimientos y la comunicación con el cliente.

**Funciones del puesto:**

- Investigar aplicaciones relacionadas con el sistema, en cuanto a funcionamiento, tecnologías y necesidades no cubiertas por estas, abarcando un análisis más genérico y uno más específico.
- Hacer un modelo lógico de alguna de estas aplicaciones.

- Definir las entidades del Sistema en desarrollo.
- Definir los actores y casos de uso del sistema.
- Elaborar los modelos de Análisis correspondientes (Dominio, User Stories).
- Llevar a cabo los análisis de factibilidad tecnológica, impacto ambiental y de riesgos.

Diseñador:

**Objetivos del puesto:** Elaborar el diseño adecuado para el sistema, a partir de los objetivos preliminares y el relevamiento previamente realizados

**Funciones del puesto:**

- Elaborar el diagrama de clases.
- Elaborar los diagramas de transición de estados correspondientes.
- Elaborar el modelo de datos (Modelo Entidad Relación).
- Diseñar las interfaces.
- Establecer cuáles serán las salidas del sistema.
- Definir las historias de usuario.

Scrum Master:

**Objetivos del puesto:** Orientar el desarrollo del nuevo producto.

**Funciones del puesto:**

- Planificar la implantación de Scrum junto con la organización.
- Ayudar a la organización a entender que interacciones con el equipo aportan valor y cuáles no.
- Asistir al Product Owner a saber cómo organizar el product backlog y maximizar el valor.
- Asegurarse de que haya una definición de done.
- Comprender y practicar la agilidad.
- Ayudar al equipo a crear productos de valor.
- Eliminar cualquier impedimento del equipo.
- Facilitar los eventos Scrum.
- Trabajar con otros Scrum Masters para aumentar la efectividad de Scrum en la organización.
- Junto con el equipo Scrum, actualizar el burndown chart.

Product owner:

**Objetivos del puesto:** Es el encargado de comunicarse con las personas interesadas en el sistema y ver qué necesidades puede solucionar el mismo.

**Funciones del puesto:**

- Encargado de hablar con las personas interesadas en el sistema.
- Establecer las prioridades de requerimientos a desarrollar.
- Comunicarse constantemente con el Scrum Master.
- Comprobar que el trabajo realizado cumpla con los requisitos de los clientes.
- Ajustar las características y prioridades por iteración.

DBA:

**Objetivos del puesto:** Implementar las bases de datos en función de los modelos realizados en la etapa de Diseño.

**Funciones del puesto:**

- Responsables del manejo, mantenimiento, desempeño y de la confiabilidad de bases de datos.
- Encargado de la mejora y diseño de nuevos modelos de las bases de datos.
- Revisar estándares y hacer que estas se cumplan.
- Conocimientos y destrezas sobresalientes en el manejo de bases de datos (Oracle, MySQL) y tecnologías SQL y ORM (JPA, Hibernate)
- Asegurar la disponibilidad de los datos 24/7.
- Monitorización continua de la base de datos.
- Mejorar la velocidad de carga de los datos.
- Realizar backup de la base de datos periódicamente.

## 2.1.5 Perfiles para los puestos de trabajo

Desarrollador Frontend:

**Título necesario:** Título terciario o Universitario correspondiente a Carreras de Sistemas o afines.

**Nivel de experiencia:** Mínimo 1 año de experiencia como front-end con tecnologías como HTML5, CSS3, JavaScript.

**Conocimientos necesarios:**

- HTML5, CSS3 y JavaScript
- jQuery.
- Bootstrap.
- Angular.
- Tener conocimientos de diseño y manejar los elementos visuales de un sitio web.
- UX (Experiencia de Usuario).

**Aptitudes y características:**

- Comunicación efectiva:
- Trabajo en equipo.
- Resolución de problemas.
- Aceptar críticas y errores.
- Buena administración del tiempo.
- Aprendizaje constante.

Desarrollador Back-End:

**Título necesario:** Título terciario o Universitario correspondiente a Carreras de Sistemas o afines.

**Nivel de experiencia:** Mínimo 1 año de experiencia como Back-End con tecnologías como Java, Spring, Hibernate, JPA.

**Conocimientos necesarios:**

- Diseño.
- Java.
- Manejo del framework Spring Boot.
- Conocimientos básicos de configuración de servidores web.

**Aptitudes y características:**

- Comunicación efectiva.
- Trabajo en equipo.
- Resolución de problemas.
- Aceptar críticas y errores.
- Buena administración del tiempo.
- Aprendizaje constante.

Ingeniero QA:

**Título necesario:** Título terciario o Universitario correspondiente a Carreras de Sistemas o afines.

**Nivel de experiencia:** 6 meses de experiencia en testing, reporte de bugs y aplicaciones de casos de pruebas.

**Conocimientos necesarios:**

- Metodologías y modelos de calidad para la industria del software.
- Metodologías de desarrollo de sistemas.
- Scrum/UML/TDD.
- CMMI, ISTQB, modelo en V, RUP.
- Herramientas para ejecución y seguimiento de pruebas.

**Aptitudes y características:**

- Dar una retroalimentación corta y clara.
- Criticar constructivamente.
- Observador.
- Meticuloso.
- Buen comunicador.
- Capacidad para ponerse en el lugar del usuario.

DevOps:

**Título necesario:** Título terciario o Universitario correspondiente a Carreras de Sistemas o afines.

**Nivel de experiencia:** 1 año de experiencia trabajando como DevOps con tecnologías como docker y kubernetes.

**Conocimientos necesarios:**

- CI/CD.
- Gitlab CI/CD.
- Selenium.
- Docker y kubernetes.
- Infraestructura de Google Cloud.

**Aptitudes y características:**

- Experiencia con herramientas específicas.
- Formación continua.
- Habilidades de Comunicación.
- Conocimiento en metodologías agile.
- Polivalencia.
- Organizar y comunicar.

Analista:

**Título necesario:** Analista en Sistemas de información o Ingeniero en Sistemas de información.

**Nivel de experiencia:** Mínimo 1 años de experiencia trabajando como analista.

**Conocimientos necesarios:**

- Nivel de idioma inglés avanzado.
- Planificación.
- Enterprise Architect.
- UML.

**Aptitudes y características:**

- Capacidad para trabajar en equipo.
- Capacidad de comunicación con el cliente.
- Creatividad al investigar y relevar.
- Actitud positiva a la hora de trabajar.
- Motivación para afrontar nuevos desafíos dentro de un proyecto de software.
- Buena presencia.

Diseñador:

**Título necesario:** Analista en Sistemas de información o Ingeniero en Sistemas de información.

**Nivel de experiencia:** Mínimo 2 años de experiencia trabajando como diseñador.

**Conocimientos necesarios:**

- Conocimientos básicos de lenguajes de programación.
- Uso en metodologías ágiles.
- Enterprise Architect.
- UML.

**Aptitudes y características:**

- Liderar con el ejemplo.
- Promover el debate y apoyarlo.
- Capacidad de comunicación con el cliente.
- Capacidad de comunicación con el equipo del proyecto.
- Creatividad al diseñar una solución
- Proactivo



- Buenos modales

Scrum Master:

**Título necesario:** Analista en Sistemas de información o Ingeniero en Sistemas de información.

**Nivel de experiencia:** Mínimo 2 años de experiencia trabajando como Scrum Master.

**Conocimientos necesarios:**

- Conocimientos básicos de lenguajes de programación.
- Uso de Microsoft Project.
- Uso en metodologías ágiles.

**Aptitudes y características:**

- Liderar con el ejemplo.
- Promover el debate y apoyarlo.
- Ser paciente y tolerante.
- Dejar que el equipo actúe.
- Autoorganizado.
- Proactivo.
- Buenos modales.

Product owner:

**Título necesario:** Título terciario o Universitario correspondiente a Carreras de Sistemas o afines.

**Nivel de experiencia:** Mínimo 1 año de experiencia trabajando como product owner.

**Conocimientos necesarios:**

- Metodologías ágiles.
- Scrum.
- Diseño de sistemas.

**Aptitudes y características:**

- Responsabilidad.
- Proactividad.
- Adaptabilidad y flexibilidad.

DBA:

**Título necesario:** Técnico en Base de Datos.

**Nivel de experiencia:** Mínimo 2 años de experiencia trabajando como administrador de base de datos, administrando cambios y actualizaciones.

**Conocimientos necesarios:**

- Modelado de datos, diseño de base de datos.
- Arquitectura del sistema.
- Instalación y configuración de motores.
- Backup y recovery.
- Conocimientos de linux.

- Bases de datos no relacionales.
- Normalización de bases de datos.

**Aptitudes y características:**

- Autoaprendizaje.
- Acostumbrado a trabajar bajo presión y en horarios no convencionales.
- Experiencia con herramientas específicas.
- Formación continua.
- Responsabilidad.
- Habilidades de Comunicación.
- Alto grado de lealtad.
- Discreción.
- Polivalencia.
- Excelentes relaciones interpersonales.

2.1.6 Estructura del equipo de trabajo

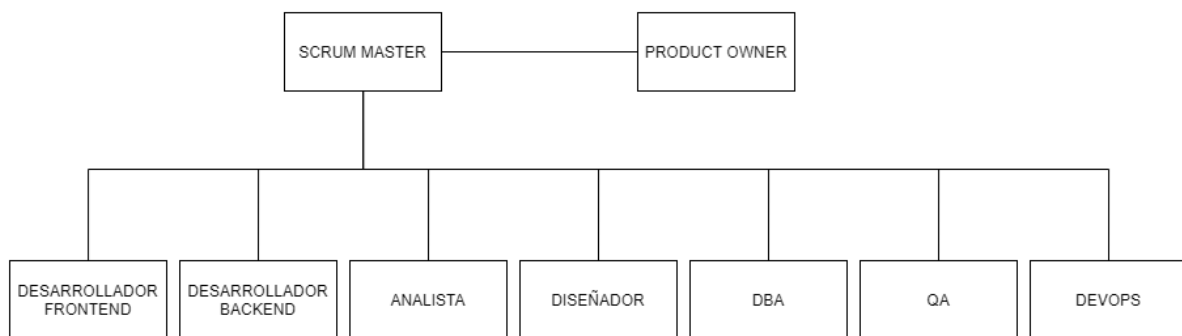


Figura 200: Estructura del equipo de trabajo

2.2 Funciones principales de los miembros del equipo

En este apartado se muestra una tabla resumen de las funciones que cumple cada miembro del equipo en las distintas etapas.

<b>Etapas/Equipo</b>	<b>Ignacio Starkloff</b>	<b>Miguel Del Corso</b>	<b>Juan Segundo Riera</b>	<b>Jose Aranciba</b>	<b>Elvio Aruta</b>
<b>Requerimientos</b>	Scrum Master	Analista	Analista	Product Owner	Analista
<b>Análisis</b>	Scrum Master	Analista	Analista	Product Owner	Analista
<b>Diseño</b>	Scrum	Diseñador	Diseñador	Product	Diseñador

	Master			Owner	
<b>Implementación</b>	Desarrollador BackEnd - Scrum Master	Desarrollador BackEnd - DBA	Desarrollador FrontEnd	Desarrollador FrontEnd - Product Owner	Desarrollador BackEnd - DevOps
<b>Pruebas</b>	Desarrollador BackEnd - Scrum Master	Desarrollador BackEnd	Desarrollador FrontEnd	Desarrollador FrontEnd - Product Owner	Ingeniero QA

## 2.3 Métodos de comunicación formal, control de avance, retroalimentación, decisiones.

### 2.3.1 WhatsApp



Figura 201: Logo de WhatsApp

WhatsApp Messenger es una aplicación de mensajería instantánea para teléfonos inteligentes, en la que se envían y reciben mensajes mediante Internet, así como imágenes, vídeos, audios, grabaciones de audio (notas de voz), documentos, ubicaciones, contactos, gifs, stickers, así como llamadas y videollamadas con varios participantes a la vez, entre otras funciones. WhatsApp se integra automáticamente a la libreta de contactos, lo que lo diferencia de otras aplicaciones, ya que no es necesario ingresar alguna contraseña o PIN para acceder al servicio.

### 2.3.2 Discord



Figura 202: Logo de discord

Discord es un servicio de mensajería instantánea freeware de chat de voz VoIP, video y chat por texto; este funciona a través de servidores; este está separado en canales ya sea de texto o de voz. Discord está disponible para Microsoft Windows, MacOS, Android, iOS y Linux. Tiene la posibilidad de ser usado desde un cliente ejecutable o también puede ser usado desde el navegador. Si bien la aplicación es de uso general, sus características la orientan hacia las comunidades de videojuegos. Desde julio de 2019, la plataforma cuenta con más de 350 millones de usuarios activos.

### 2.3.2 Google drive



Figura 203: Logo de Google drive

Google Drive es un servicio de alojamiento de archivos que fue introducido por la empresa estadounidense Google el 24 de abril de 2012.

Cada usuario cuenta con 15 gigabytes (GB) de espacio gratuito para almacenar sus archivos, ampliables mediante diferentes planes de pago. Es accesible a través del sitio web desde computadoras y disponen de aplicaciones para Android e iOS que permiten editar documentos y hojas de cálculo

### 2.3.3 Moodle



Figura 204: Logo de moodle

Moodle es una herramienta de gestión de aprendizaje (LMS), o más concretamente de Learning Content Management (LCMS), de distribución libre, escrita en PHP. Está concebida para ayudar a los docentes a crear comunidades de aprendizaje en línea, Moodle es usada en blended learning, educación a distancia, clase invertida y diversos proyectos de e-learning en escuelas, universidades, oficinas y otros sectores. La versión más reciente es la 3.9.2+.

### 2.3.4 Trello



Figura 205: Logo de Trello

Trello es un software de administración de proyectos con interfaz web y con cliente para iOS y Android para organizar proyectos. Se caracteriza por ser una solución fácil, gratuita, flexible y visual para gestionar proyectos y organizarlo todo.

## 2.4 Gestión de Configuración del Software: Método de gestión de versionado durante todo el proyecto.

### 2.4.1 Gestión de configuración del software



Figura 206: Logo de git

Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando estas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora incluyendo coordinar el trabajo que varias personas realizan sobre archivos compartidos en un repositorio de código.

Al principio, Git se pensó como un motor de bajo nivel, sobre el cual otros pudieran escribir la interfaz de usuario o front end como Cogito o StGIT. 3. Sin embargo, Git se ha convertido

desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux.

## 2.4.2 Formato del versionado

Para el versionado utilizaremos 3 números: X.Y.Z. El cual cada uno indica una cosa diferente:

El primero (X) se le conoce como versión mayor y nos indica la versión principal del software.

El segundo (Y) se le conoce como versión menor y nos indica nuevas funcionalidades.

El tercero (Z) se le conoce como revisión y nos indica que se hizo una revisión del código por algún fallo.

**4.2.1**  
MAJOR *Minor* patch

Figura 207: Formato del versionado

## Capítulo III: Factibilidad

### 3.1 Definición y descripción de recursos para cada una de las actividades.

Actividad	Recurso
Definición del sistema	Product Owner Scrum Master
<b>Relevamiento General</b>	
Relevamiento de las organizaciones	Analista 1 Analista 2 Analista 3 Scrum Master
Funciones detectadas	Analista 1 Diseñador 1
Tecnologías de Información	DevOps Desarrollador Back-End 1 Desarrollador Front-End 1
<b>Relevamiento detallado y análisis del Sistema</b>	
Detalle, explicación y documentación detallada de todas las funciones seleccionadas	Desarrollador Back-End 1 Desarrollador Front-End 1 Scrum Master Product owner DevOps
Modelo lógico del sistema	Analista 3 Diseñador 1 Diseñador 2 Desarrollador Back-End 1 Desarrollador Front-End 1
Problemas y necesidades detectados en las funciones relevadas en detalle y en su entorno organizacional	Scrum Master Product owner Analista 2
Objetivos y alcances preliminares del nuevo Sistema	Product owner Scrum Master Diseñador 2
<b>Planificación</b>	
Estimación de actividades	Product Owner

	Scrum Master Desarrollador BackEnd 1 Desarrollador FrontEnd 1 Desarrollador BackEnd 2 Desarrollador FrontEnd 2
Selección de metodologías	Product Owner Scrum Master
Definición del equipo de trabajo	Product Owner Scrum Master
Definición de tecnologías	Desarrollador BackEnd 1 Desarrollador FrontEnd 1
Definición de herramientas de comunicación	Scrum Master Product Owner
Definición de herramientas para el seguimiento de tareas	Scrum Master Product Owner
Definición de herramientas para la gestión de configuración del software	Scrum Master Desarrollador BackEnd 1 Desarrollador FrontEnd 1
Definición y descripción de recursos para cada una de las actividades	Product Owner Scrum Master Desarrollador BackEnd 1 Desarrollador FrontEnd 1
Realizar estudio de factibilidad	Product Owner
Costos de infraestructura y tecnologías	Product Owner Scrum Master
Análisis de riesgos	Product Owner Scrum Master Desarrollador Back-End 1 Desarrollador Front-End 1 DevOps
Análisis de impacto ambiental	Analista 2 Product owner Scrum Master
<b>Diseño</b>	
Objetivos y alcances definitivos del nuevo Sistema	Product Owner Scrum Master Desarrollador Back-End 3 Desarrollador Front-End 2



Salidas del Sistema	Product Owner Scrum Master
MockUps	Desarrollador FrontEnd 1 Desarrollador FrontEnd 2
Modelo de datos	Desarrollador Back-End 1 Desarrollador Back-End 2 Desarrollador Front-End 1 Diseñador 1 DevOps
Definición de casos de uso	Product Owner Diseñador 2
Definición de historias de usuario	Product Owner Desarrollador Back-End 1 Desarrollador Front-End 1
Estimación de historias de usuario	Product Owner Scrum Master
Definición de Ceremonias de la metodología ágil	Scrum Master
Diseño de módulo seguridad	Desarrollador BackEnd 1 Desarrollador Front-End 1
Diseño de módulo eventos	Desarrollador BackEnd 2 Desarrollador Front-End 1
Diseño de módulo reportes	Desarrollador BackEnd 3 Desarrollador Front-End 1
Diseño de modulo publicidad	Desarrollador BackEnd 1 Desarrollador Front-End 1
Diseño de modulo inscripcion	Desarrollador BackEnd 2 Desarrollador Front-End 1
Diseño de módulo pagos	Desarrollador BackEnd 3 Desarrollador Front-End 2
Diseño de módulo feedback	Desarrollador BackEnd 1 Desarrollador Front-End 2
Diseño de módulo parametrización	Desarrollador BackEnd 2 Desarrollador Front-End 2
Diseño de módulo perfil	Desarrollador BackEnd 2 Desarrollador Back-End 1
Diseño de pruebas de integración	Desarrollador BackEnd 3

	Desarrollador Back-End 1
Diseño de pruebas de seguridad	Desarrollador BackEnd 1 Desarrollador Back-End 1
Diseño de pruebas de stress/carga	Desarrollador BackEnd 2 Desarrollador Front-End 1
<b>Desarrollo e implementación</b>	
Desarrollo del formato sobre el cual vamos a trabajar	Scrum Master DevOps
Programación, pruebas y documentación del módulo seguridad	Desarrollador BackEnd 2 Desarrollador FrontEnd 2 Ingeniero QA
Programación, pruebas y documentación del módulo eventos	Desarrollador BackEnd 3 Desarrollador FrontEnd 3 Ingeniero QA
Programación, pruebas y documentación del módulo reportes	Desarrollador BackEnd 1 Desarrollador FrontEnd 1 Ingeniero QA
Programación, pruebas y documentación del modulo publicidad	Desarrollador BackEnd 2 Desarrollador FrontEnd 2 Ingeniero QA
Programación, pruebas y documentación del módulo inscripción	Desarrollador BackEnd 3 Desarrollador FrontEnd 3 Ingeniero QA
Programación, pruebas y documentación del módulo pagos	Desarrollador BackEnd 2 Desarrollador FrontEnd 2 Ingeniero QA
Programación, pruebas y documentación del módulo feedback	Desarrollador BackEnd 1 Desarrollador FrontEnd 1 Ingeniero QA
Programación, pruebas y documentación del módulo parametrización	Desarrollador BackEnd 1 Desarrollador FrontEnd 1 Ingeniero QA
Programación, pruebas y documentación del módulo perfil	Desarrollador BackEnd 1 Desarrollador FrontEnd 1 Ingeniero QA
Realizar el manual de usuario del Sistema completo	Desarrollador BackEnd 2 Desarrollador FrontEnd 2 Ingeniero QA

Realizar la implementación del Sistema	Desarrollador BackEnd 3 Desarrollador FrontEnd 3 Ingeniero QA
Realizar capacitación a usuarios	Diseñador 1 Diseñador 2 Desarrollador Back-End 1 Desarrollador Front-End 1 Ingeniero QA
Demo del sistema y ensayo de exposición	Scrum Master Desarrollador Back-End 1 Desarrollador Back-End 2 Desarrollador Front-End 1
Exposición anual de Proyecto de Sistemas	Scrum Master Desarrollador Back-End 1 Desarrollador Back-End 2 Desarrollador Back-End 3 Desarrollador Front-End 1 Desarrollador Front-End 2

### 3.2 Diagrama de recursos

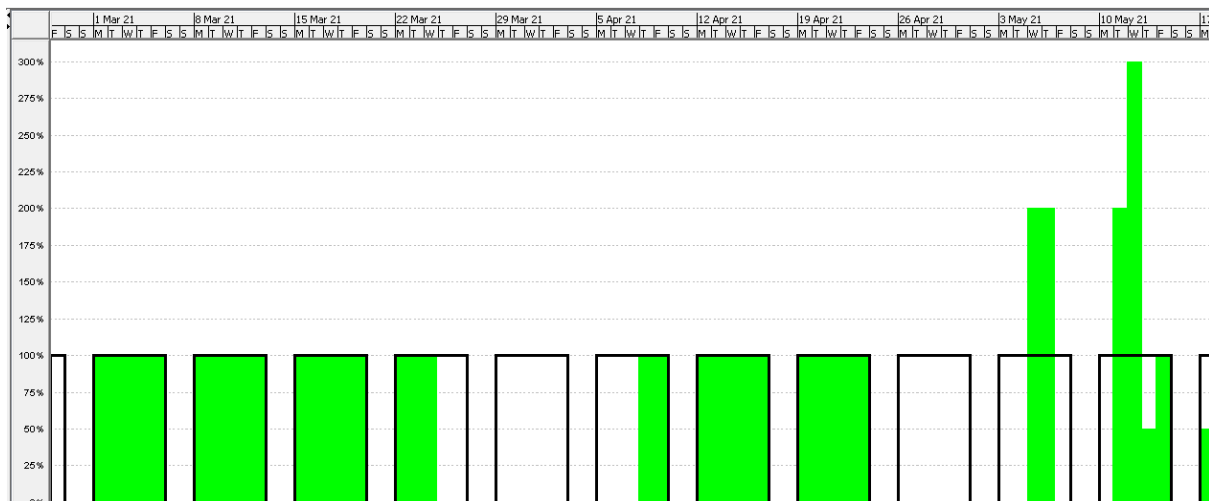


Figura 208: Diagrama de recursos del Scrum Master

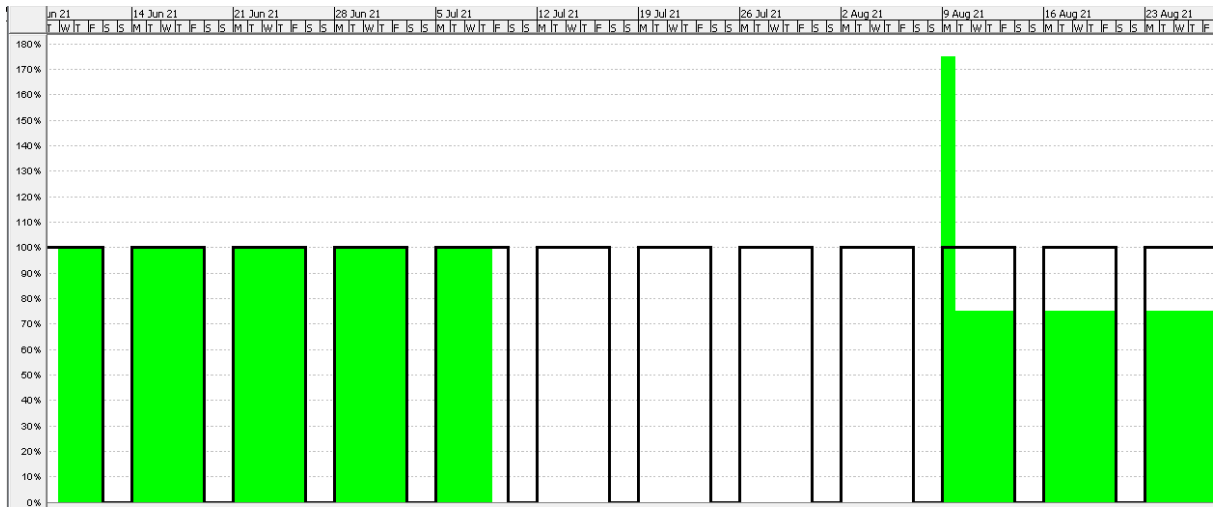


Figura 209: Diagrama de recursos del desarrollador BackEnd 1

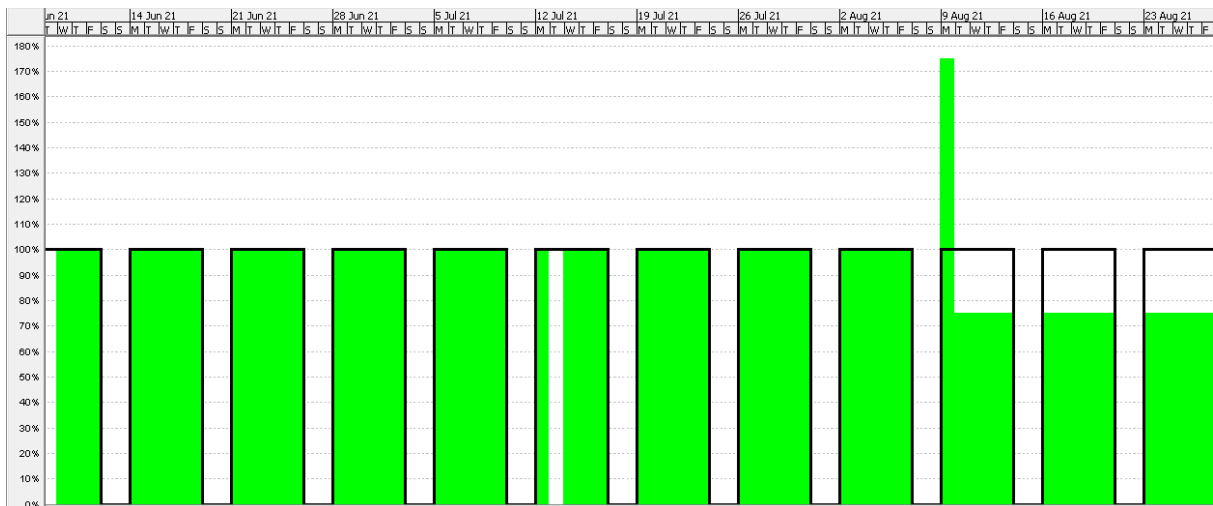


Figura 210: Diagrama de recursos del desarrollador FrontEnd 1

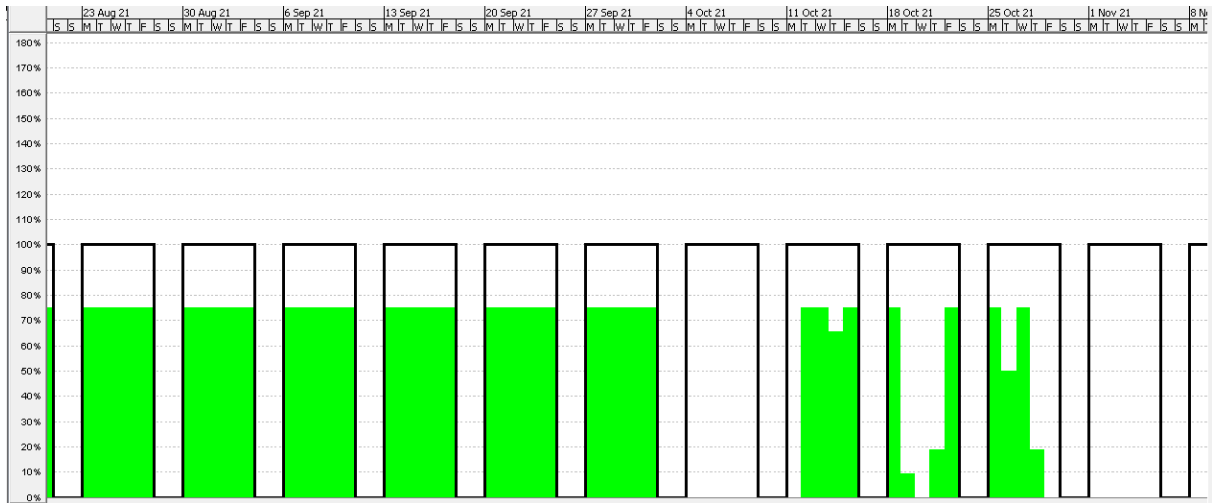


Figura 211: Diagrama de recursos del desarrollador BackEnd 2

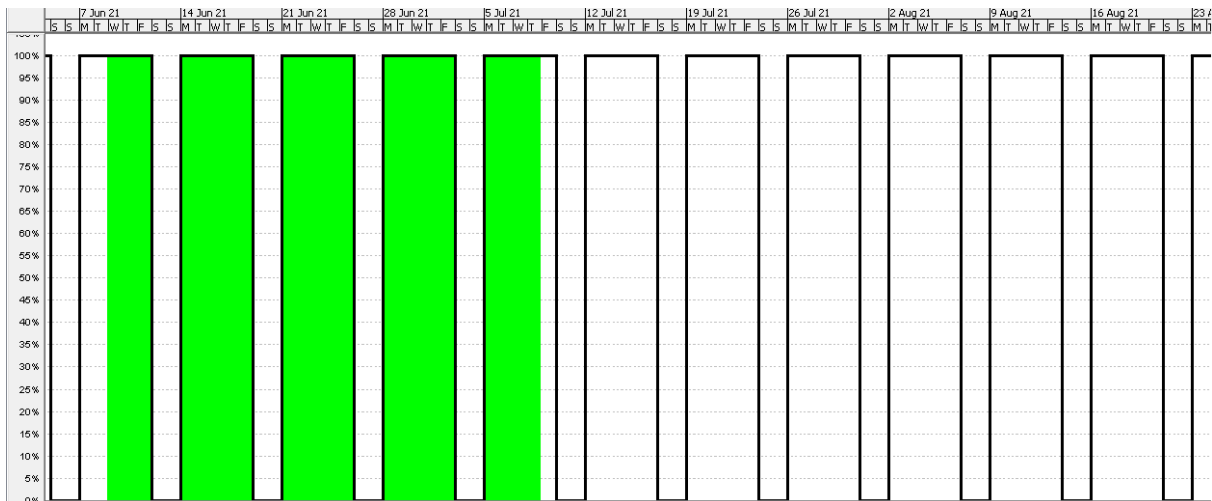


Figura 212: Diagrama de recursos del desarrollador FrontEnd 2

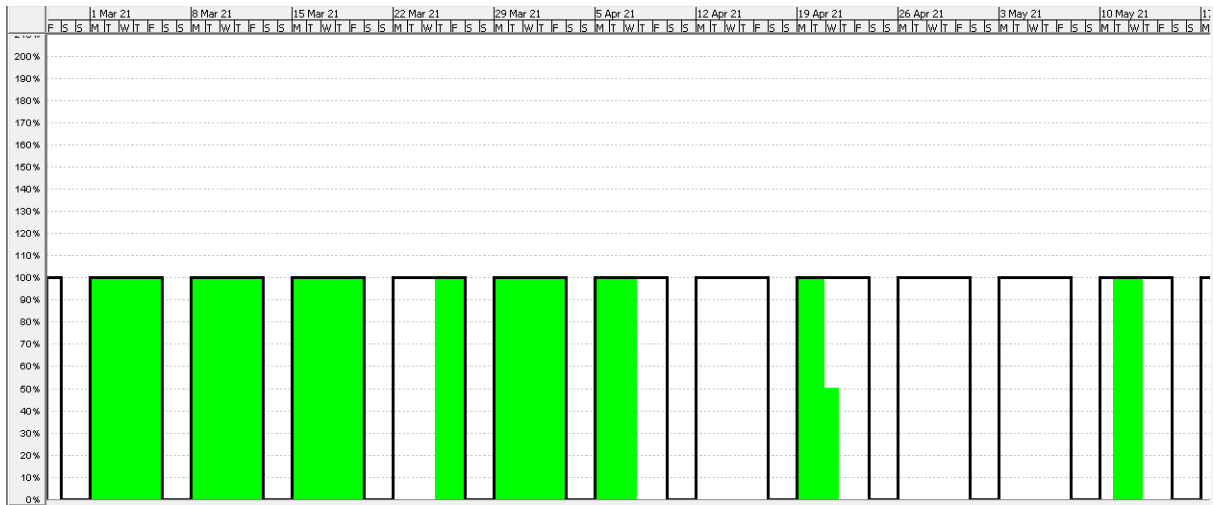


Figura 213: Diagrama de recursos del analista 1

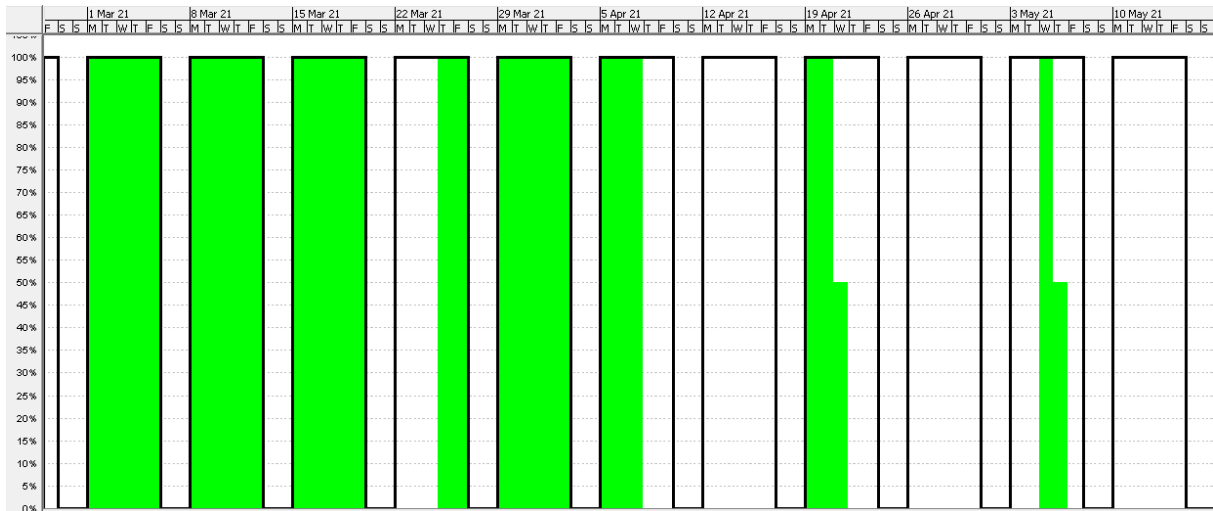


Figura 214: Diagrama de recursos del analista 2

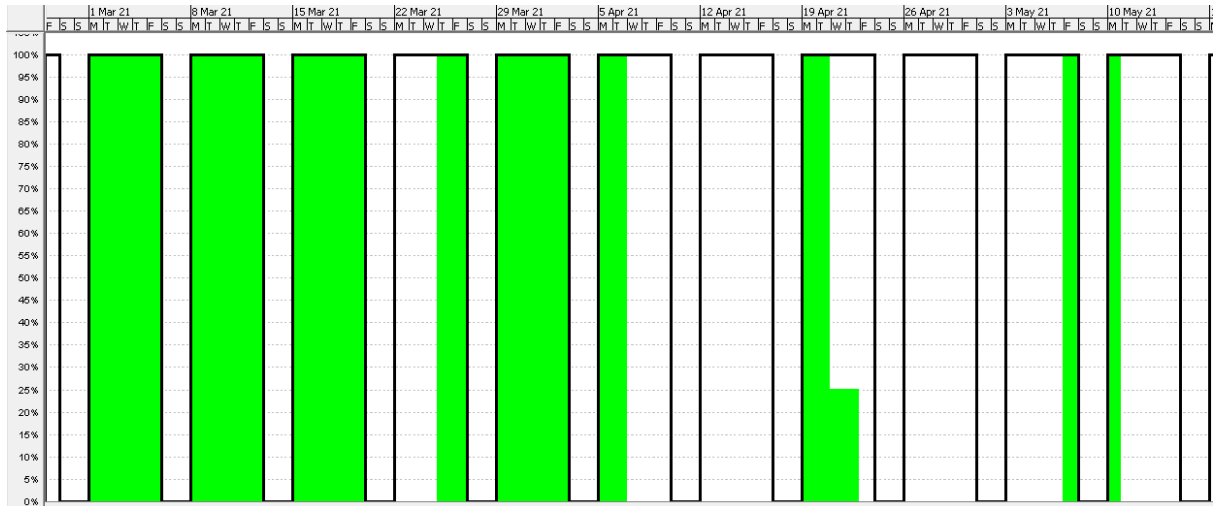


Figura 215: Diagrama de recursos del analista 3

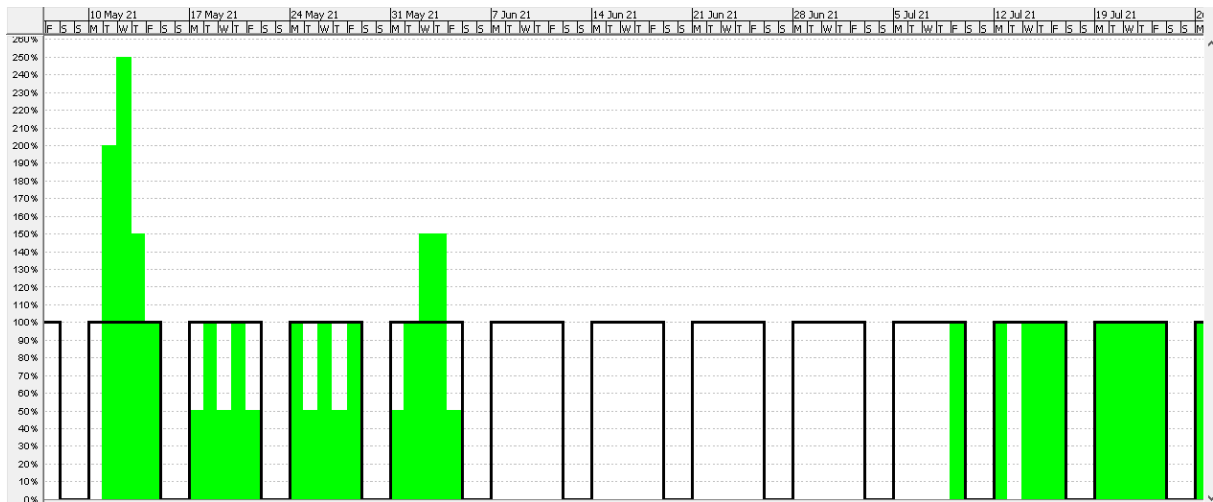


Figura 216: Diagrama de recursos del diseñador 1

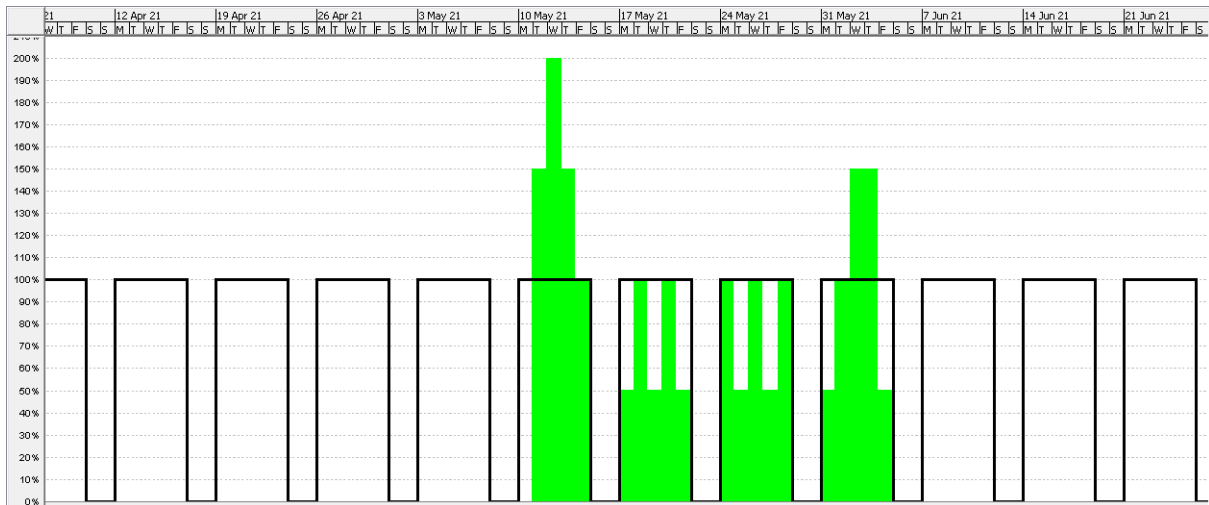


Figura 217: Diagrama de recursos del desarrollador diseñador 2

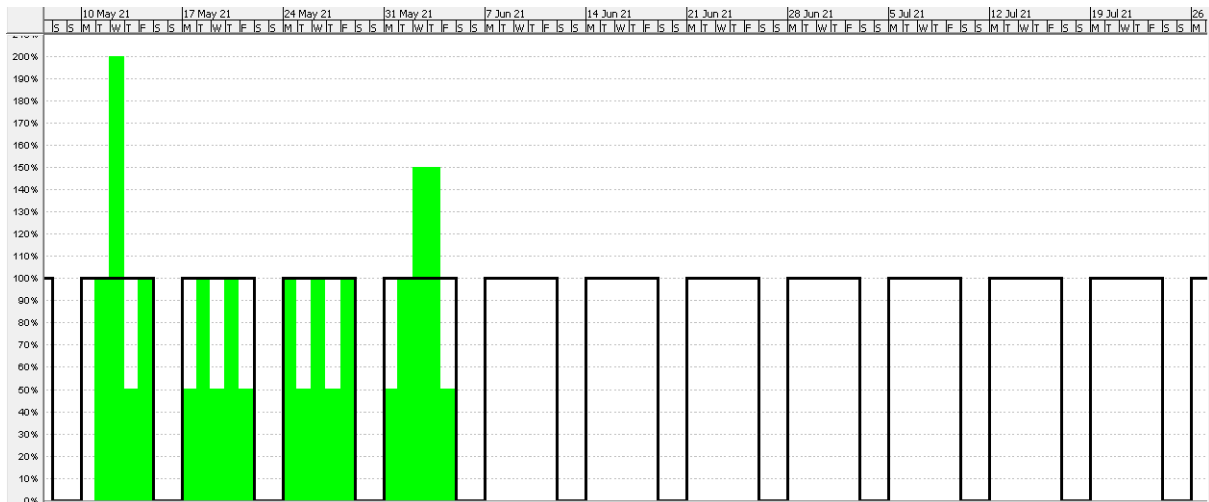


Figura 218: Diagrama de recursos del diseñador 3

### 3.3 Análisis de factibilidad

Un estudio de factibilidad es el que se realiza para determinar la posibilidad de poder desarrollar el proyecto. Vamos a realizar los distintos tipos de factibilidad:

#### 3.3.1 Factibilidad Técnica

Los aspectos que analizamos son:

**Volumen de datos, tipos de datos:** El sistema a lo largo del tiempo va a manejar un gran volumen de datos, ya que por cada torneo vamos a tener guardar mucha información relacionada con la misma (por ejemplo las distintas jornadas que tiene el torneo). Además, que en un futuro se planean agregar más deportes por lo que la cantidad de datos



almacenados aumentará. La cantidad de datos almacenados por cada deporte está cerca del Gigabyte.

**Tipo de procesamiento de datos:** Con toda la información que dispone el sistema, se realizan reportes, estadísticas, resúmenes.

**Nivel de automatización de las funciones:** Las principales funciones que el sistema efectúa, están automatizadas, con la excepción de la confirmación del deportista al evento, que si requiere intervención del usuario.

**Tipos de captura de datos:** El sistema va a capturar la mayoría de los datos por formularios web, las imágenes y videos van a ser capturadas a través de un componente específicamente diseñado para tal fin, con un límite máximo de 10 MB para las imágenes y 300 MB para los videos.

**Frecuencia y volumen de ingreso de datos:** La frecuencia y volumen de ingreso de datos depende principalmente de la cantidad de eventos registrados. Estimamos que se van a ejecutar 5 registros de eventos por día. Los días que hay eventos hay una gran actualización de datos, por ejemplo del marcador del partido, además de los comentarios y fotos de los participantes.

**Formularios** Son la principal fuente de ingresos de datos al sistema realizada por los usuarios finales.

**Backups:** Se estima que se van a llevar a cabo backups incrementales todos los días, y además cada 1 mes se procederá un backup completo.

**Seguridad:** El acceso a los datos del sistema estará expuesto únicamente a través de una API, la cual necesita de una clave especial. Para evitar bots, ataques DDoS se decide contratar el servicio gratuito de CloudFlare.

**Recuperación:** En caso de haber un problema con la base de datos se tomará el backup del día anterior, perdiendo los datos del día solamente.

**Integración con otros Sistemas y otras TI internas y externas:** El sistema se encuentra integrado con una API de mapas para poder determinar la ubicación del evento. También se hace una API del clima para poder determinar el clima que habrá cuando se haga el evento.

### **Crecimiento estimado en la empresa que afecte al Sistema**

**Crecimiento funcional y de TI para el Sistema:** A medida que vayan aumentando los usuarios, se necesitará incrementar la cantidad de personas trabajando para poder dar soporte a los mismos. También se tendrá que realizar cambios en los planes contratados para almacenar el sitio web y la base de datos.

**Flexibilidad para nuevas TI** Con las tecnologías seleccionadas para el desarrollo del sistema sería bastante fácil expandir o cambiar el sistema con nuevas tecnologías. Se detecta la necesidad de añadir nuevos métodos de pago como transferencia bancaria o bitcoin, que por el momento quedan fuera del alcance inicial del sistema.

Ventajas y desventajas de las tecnologías utilizadas en el proyecto

GIT



Figura 219: Figura de git

Ventajas

- Es gratis de utilizar.
- Seguimiento de errores.
- Es muy rápido.
- Puede manejar grandes proyectos.
- GitHub es fácil de usar.
- Cuenta con una potente comunidad de desarrolladores en todo el mundo.
- Permite regresar a versiones anteriores de forma sencilla y muy rápida.

Desventajas

- La documentación es complicada.
- Tienen una interfaz gráfica de usuario y una facilidad de uso deficientes.
- No proporcionan mecanismos de control de acceso en caso de seguridad.

MONGODB



Figura 220: Figura de mongoDB

Ventajas

- Es gratis de utilizar.
- Escalabilidad.
- Tiene una gran documentación.
- Complemento perfecto para JavaScript.

Desventajas

- No es una base de datos adecuada para aplicaciones con transacciones complejas.

- Es una tecnología joven.
- No tiene Joins para consultas.

## ANGULAR



Figura 221: Figura de angular

### Ventajas

- Cuenta con un gran enlace de datos MVC que permite que el desarrollo de las aplicaciones sea rápido.
- El uso de HTML como lenguaje declarativo hace que sea muy intuitivo.
- Es una solución integral para un rápido desarrollo front-end pues no necesita otro marco o complemento.
- El soporte completo de TypeScript ayuda a la creación de aplicaciones de mayor volumen.
- Las pruebas son sencillas de escribir.

### Desventajas

- Es grande y complicado debido a las múltiples formas de hacer lo mismo.
- Las implementaciones no siempre escalan bien.
- Es lento cuando se muestran grandes cantidades de datos.

## NODEJS



Figura 222: Figura de nodeJS

### Ventajas

- Escalabilidad de manera sencilla.
- Rendimiento.
- NPM.
- Javascript.
- Popularidad.

### Desventajas

- No es adecuado para aplicaciones intensivas de CPU.
- El ecosistema del paquete es una preocupación constante.
- API Inestable: La API de Node tiene la mala costumbre de cambiar en formas que rompen la compatibilidad hacia atrás de versión en versión.

- No está Probado lo Suficiente.

## KUBERNETES



Figura 223: Figura de kubernetes

### Ventajas

- Escalabilidad y balanceo de carga.
- Aislamiento de procesos y aplicaciones.
- Facilidades para el despliegue.
- Optimización de recursos automáticos recursos de forma automática.
- Alta disponibilidad.

### Desventajas

- Kubernetes es muy complejo y puede reducir la productividad.
- Kubernetes puede ser una exageración para aplicaciones simples.
- La transición a Kubernetes puede ser engorrosa.
- Kubernetes puede ser más caro que sus alternativas.

## DOCKER



Figura 224: Figura de docker

### Ventajas

- Se obtiene una mayor consistencia entre los entornos de prueba y los entornos de producción.
- Se obtiene mayor modularidad.
- Aislamiento y seguridad.
- Facilita el testing.
- Facilita el despliegue de aplicaciones al entorno de producción.

### Desventajas

- Se requiere mínimo la versión de Kernel 3.8.
- Solo soporta sistemas operativos Linux de arquitectura de 64 bits.
- Para Windows aún se encuentra en fase de desarrollo.
- No todas las aplicaciones son compatibles o recomendables de usar con docker.
- Infraestructura compartida.

## WHATSAPP



Figura 225: Figura de WhatsApp

### Ventajas

- Gratuita.
- Sin publicidad.
- Cifrado extremo a extremo.
- Fácil de compartir contenido multimedia.
- Muy popular.
- Multiplataforma.
- Inmediatez.

### Desventajas

- Requiere ser actualizada cada cierto tiempo.
- Requiere compartir el número de teléfono.
- No se puede utilizar con personas que no tienen WhatsApp.
- Puede ocupar mucho espacio.
- No funciona sin Internet.

## DISCORD



Figura 226: Figura de discord

### Ventajas

- Debido a su funcionamiento optimizado, permite una comunicación eficiente en redes de baja latencia y conexión lenta.
- Invita a los usuarios a crear bots para automatizar tareas y simplificar ciertas rutinas al momento de crear sesiones de juego.
- Cuenta con numerosas opciones de personalización. Esto permite tener una experiencia cómoda y que se adapta a las necesidades particulares de un usuario o grupo de usuarios.
- Sincronización con otras plataformas.
- Facilidad de uso.
- Comunidades oficiales.

### Desventajas

- El sistema de videollamadas no se ha terminado de optimizar, por lo que es recurrente ver fallas e interrupciones en la comunicación.
- Algunos contenidos limitados por pago

## GOOGLE DRIVE



Figura 227: Figura de Google Drive

### Ventajas

- Múltiples usuarios pueden editar los mismos archivos al mismo tiempo
- El servicio de Google Drive elimina la preocupación de modificar un archivo irrevocablemente, dado que almacena también antiguas versiones de los documentos.
- Cuenta con herramientas de administración centralizadas, a las que se accede desde el panel de control, que facilitan las tareas de añadir o eliminar almacenamiento para usuarios específicos o para grupos de ellos. Mientras tanto, los archivos se transmiten de forma cifrada entre el navegador del usuario y el servidor, aumentando la seguridad de los datos transferidos.
- No necesita ningún tipo de mantenimiento.
- Almacenamiento escalable.

### Desventajas

- Las personas pueden observar las páginas que utilizaste.
- Si el documento tiene varios editores o colaboradores pudiera ser eliminado por uno de ellos.
- Es obligatorio el uso de internet.
- Para acceder a Google Drive necesitas una cuenta de gmail

## MOODLE



Figura 228: Figura de moodle

### Ventajas

- Facilita la comunicación de los docentes y estudiantes fuera del horario de clases.
- Ayuda al aprendizaje cooperativo, ya que permite la comunicación a distancia mediante foros, correo y chat.

- Posibilidad de diversos métodos de evaluación y calificación.
- Creación de cursos virtuales y entornos de aprendizaje virtuales.
- Se encuentra traducido a más de 70 idiomas.
- Los recursos que el docente entrega a sus estudiantes pueden ser de cualquier fuente y con cualquier formato.
- Lleva registro de acceso de los estudiantes y un historial de las actividades de cada estudiante.

#### Desventajas

- Se da una sensación de aislamiento por parte de los estudiantes.
- El seguimiento a cada estudiante se dificulta para el docente por la forma de impartir el aprendizaje.
- Se actualiza muy fácilmente desde una versión anterior a la siguiente.
- Muestra los mismos contenidos a todos los estudiantes.
- La comunicación y colaboración constante es importante, puesto que no hay la presencia del docente.
- Un fallo en los servidores o caída del servicio de internet puede dejar al usuario inhabilitado para realizar sus actividades.

#### TRELLO



Figura 229: Figura de trello

#### Ventajas

- Es rápido y práctico.
- Permite agregar comentarios.
- Las tareas y proyectos son compartidos en tiempo real.
- Se pueden crear mucho boards.
- Aplicación web.
- Permite que todos los integrantes puedan participar.
- Trabajamos más organizados (Equipo).

#### Desventajas

- No te llegan notificaciones de las actualizaciones.
- Es necesario un tiempo prudencial para adecuarse al entorno y entender las diferentes funciones que nos encontramos.
- No cuenta con etiquetas globales de proyecto, sino que en cada tablero se deben establecer las etiquetas.
- No muestra dependencias de tareas.
- La comunicación no es su fuerte.
- No es una herramienta completa de PM.

## Conclusiones de factibilidad técnica

Se concluye que el proyecto es factible técnicamente, ya que se dispone de los conocimientos, habilidades, equipos y herramientas necesarias para llevar a cabo los módulos involucrados en el proyecto.

### 3.3.2 Factibilidad Operativa

La factibilidad hace referencia a todos aquellos recursos que intervienen en algún tipo de actividad del proyecto.

**Personas (PPPP), personal permanente, temporario, asesoramiento:** Por el momento solamente vamos a participar en el proyecto los 5 miembros del equipo, no vamos a tener ningún tipo de personal permanente, temporario o asesoramiento de ningún tipo.

#### Capacitación del equipo:

Para determinar si el proyecto es factible desde el punto de vista operativo los miembros del equipo deberán estar capacitados en las siguientes tecnologías:

- GIT
- MongoDB
- Angular
- NodeJS
- Kubernetes
- Docker

Hay al menos 1 miembro del equipo que conoce bien cada una de las tecnologías mencionadas, además se realizarán capacitaciones para los miembros cuyo nivel sea muy bajo o nulo.

La siguiente tabla muestra una media del nivel de conocimiento actual por cada una de las tecnologías:

Tecnología	Conocimientos / Experiencia
Git	MEDIA
MongoDB	BAJO
Angular	ALTO
NodeJS	MEDIA
Kubernetes	MEDIA



Docker	MEDIA
--------	-------

**Recursos necesarios:**

Se realiza un análisis de los recursos mínimos que son necesarios para llevar a cabo el proyecto. Los recursos detectados son:

- Conexión a internet con una velocidad mínima de 3 MB.
- Navegador web moderno.

Respecto del hardware como mínimo se necesitará:

Hardware	Características
4 Notebooks o computadoras	<b>Procesador:</b> i3 o superior <b>RAM:</b> 4 GB o superior <b>Disco:</b> 1 TB y preferentemente un SSD <b>Sistema Operativo:</b> Windows 10 o distribución linux basada en debian.
4 teclados y mouses (solo en caso de computadora)	—
4 monitores	Como mínimo un tamaño de pantalla de 15" y una resolución de 1366x768

Todos los integrantes del equipo poseen los recursos necesarios tanto de hardware como de internet.

**Resolución de conflictos:** En caso de que se produzca un conflicto el scrum master será el encargado de resolver el problema de la mejor manera posible, es decir de una forma objetiva y sin preferencias. Se va a tratar de que en caso de hacer una crítica a algunos de los miembros del equipo, esta sea constructiva para así evitar conflictos.

**Motivación, rendimiento, cumplimiento, rechazo al Sistema:** El scrum master será el encargado de mantener la motivación del equipo lo más alta posible, ya que si la motivación es alta el rendimiento y la productividad aumenta.

Respecto al **mantenimiento**, por parte del software se realizará un mantenimiento de manera constante durante 6 meses, luego se realizará mediante la detección de bugs por parte de los usuarios de la aplicación. Respecto al hardware no se hará ningún tipo de mantenimiento, ya toda la aplicación se encuentra en la nube.

Para efectuar el **testing** se tendrá a un tester en el equipo que ejecutará las pruebas unitarias, de integración y de seguridad para reducir de manera significativa la cantidad de errores que llegarán a producción y verán los usuarios finales. Sin embargo, si los usuarios llegan a encontrar problemas en el sistema, tienen la posibilidad de reportar el error al

administrador y el mismo se tratará de arreglar lo más rápido posible y se realizarán pruebas de regresión antes de volver a subir la aplicación.

#### Conclusiones de factibilidad operativa

Concluimos que el proyecto es factible operativamente, ya que se cuenta con los materiales de hardware requeridos. El conocimiento de los miembros del equipo es suficiente para llevar a cabo el desarrollo del sistema (previa capacitación).

### 3.3.3 Factibilidad Legal

A continuación se hace un análisis de la factibilidad legal del proyecto

#### **Aspectos importantes a tener en cuenta:**

##### **Licencias**

Para evitar cualquier tipo de riesgo legal, previo al lanzamiento de la aplicación se adquirirán las licencias necesarias de los diferentes softwares y frameworks utilizados para crearla.

##### **Ley Orgánica de Protección de Datos (LOPD)**

Es una ley que tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor, intimidad y privacidad personal y familiar.

Su objetivo principal es regular el tratamiento de los datos y ficheros, de carácter personal, independientemente del soporte en el cual sean tratados, los derechos de los ciudadanos sobre ellos y las obligaciones de aquellos que los generan o tratan.

##### **Seguridad de la información del sistema:**

La información proporcionada por la aplicación debe mantenerse encriptada en todo tipo de comunicación y solo deberán tener acceso a la información correspondiente según el rol que presenten frente al sistema al identificarse con su usuario y contraseña.

#### Conclusiones de la factibilidad legal

Podemos concluir que el proyecto es factible desde el punto de vista de la legalidad.

### 3.3.4 Factibilidad Económica

#### 3.3.4.1. Costos económicos generales iniciales

Recursos	Costos
Costos de personal	ARS \$2.071.480,00
Costos extras	ARS \$25.000,00
<b>Total</b>	<b>ARS \$2.096.480,00</b>

\*Se estiman los costos extras en cuanto a riesgos de estropeo de equipos o infraestructura de redes.

#### Retorno de inversión

Se evalúa luego, el retorno de inversión mensual considerando los ingresos de :

- **Ventas de versiones pagas:** ARS \$1.500.000,00 anuales considerando una media de 1000 personas durante todo el año por mes. \$ 125.000 mensuales
- **Ganancias en publicidad de empresas externas:** un ingreso anual de ARS \$500.000,00 con un promedio de tráfico de 5.000 personas solo en la página web

#### Cálculo del retorno:

(valor del beneficio - valor total de la inversión)/valor de la inversión

ANUAL primer año.(con costos iniciales):

$$ROI_A = \text{ARS } \$2.000.000,00 - \text{ARS } \$2.096.480,00 / \text{ARS } \$2.387.320,00 = -0,04602$$

Como podemos observar el ROI en el primer año nos dio negativo. Hay que aclarar que durante el primer año nos pareció importante priorizar los activos intangibles, como el posicionamiento social para estar bien arraigados en el mercado y a partir del segundo año si bien se planea seguir dándole importancia a los beneficios económicos intangibles también se empezará a buscar obtener beneficios económicos tangibles. Teniendo en cuenta que a partir del segundo año no se desarrollará tanto, sino que se dará mantenimiento, los costos de sueldos se reducirán y el ROI será positivo.

### 3.3.5. Beneficios económicos intangibles

- **Fidelización con el cliente:** Los servicios posventa y la inversión en servicios de atención al cliente son necesarios para mejorar la fidelización del cliente, asegurando que estos no dejarán de utilizar nuestros servicios y nos recomendarán.

- **Experiencia profesional:** Uno de los beneficios a la hora de realizar el sistema es la experiencia obtenida por parte de los miembros del grupo en las distintas tecnologías utilizadas. De este factor, no se obtienen beneficios económicos.
- Incremento de la **satisfacción de los deportistas** debido a la utilización de la aplicación.
- Posicionamiento social para futuros proyectos.

### 3.3.6. Costos económicos intangibles

- Estrés generado por hacer un informe de proyecto final
- **Baja innovación:** Esto es debido a no ser pioneros elaborando sistemas de la índole de administración de gerencia miento de eventos deportivos.
- **Pérdida de reputación:** Debido al hecho de fundar la empresa en un país como Argentina que se posiciona debajo de la media en cuanto a la mayoría de los factores económicos y emprendedurismo.

#### Conclusiones de la factibilidad económica

Concluimos que el proyecto tiene un ROI bajo en el inicio del mismo debido a los costos iniciales necesarios para lanzar el proyecto y que los beneficios iniciales son muy bajos tomando en cuenta la baja cantidad de usuarios que conozcan y empleen la aplicación.

## 3.4 Costos desagregados por recursos (personal, tecnología) con periodicidad mensual

### Costos de personal

El análisis de costos del personal es el análisis más importante cuando se habla de costos en un proyecto de software, ya que la mayor parte de los costos se refieren a la mano de obra de los desarrolladores e integrantes del equipo que intervienen en el proyecto.

El costo de cada rol dentro del equipo se estimaron basándonos en la plataforma [openqube.io](https://openqube.io),

reconocida plataforma donde se encuestan a profesionales del rubro tecnológico. (Los salarios son actualizados a enero 2021).

Además, tendremos en cuenta los salarios para un nivel de experiencia de cada profesional equivalente a un *Semi Senior*.

*También, se ha de considerar que el salario puede variar según los estudios formales de cada integrante del equipo.*

Decidimos que los costos de personal van a estar desagregados por mes:

<b>Marzo</b>			
<b>Nombre del recurso</b>	<b>Horas</b>	<b>\$ / hora</b>	<b>Sub total</b>
Scrum Master	144	ARS \$380,00	ARS \$54.720,00
Product owner	144	ARS \$400,00	ARS \$57.600,00
Analista 1	160	ARS \$300,00	ARS \$48.000,00
Analista 2	160	ARS \$300,00	ARS \$48.000,00
Analista 3	160	ARS \$300,00	ARS \$48.000,00
<b>Total</b>			<b>ARS \$256.320,00</b>

<b>Abril</b>			
<b>Nombre del recurso</b>	<b>Horas</b>	<b>\$ / hora</b>	<b>Sub total</b>
Scrum Master	96	ARS \$380,00	ARS \$36.480,00
Product owner	16	ARS \$400,00	ARS \$6.400,00
Analista 1	60	ARS \$300,00	ARS \$18.000,00
Analista 2	60	ARS \$300,00	ARS \$18.000,00
Analista 3	52	ARS \$300,00	ARS \$15.600,00
<b>Total</b>			<b>ARS \$94.480,00</b>

<b>Mayo</b>			
<b>Nombre del recurso</b>	<b>Horas</b>	<b>\$ / hora</b>	<b>Sub total</b>
Scrum Master	148	ARS \$380,00	ARS \$56.240,00
Product owner	124	ARS \$400,00	ARS \$49.600,00
QA	32	ARS \$325,00	ARS \$10.400,00
Analista 1	16	ARS \$300,00	ARS \$4.800,00
Analista 2	12	ARS \$300,00	ARS \$3.600,00
Analista 3	16	ARS \$300,00	ARS \$4.800,00
Diseñador 1	120	ARS \$320,00	ARS \$38.400,00
Diseñador 2	112	ARS \$320,00	ARS \$35.840,00
Diseñador 3	100	ARS \$320,00	ARS \$32.000,00
<b>Total</b>			<b>ARS \$235.680,00</b>

Junio			
Nombre del recurso	Horas	\$ / hora	Sub total
Scrum Master	36	ARS \$380,00	ARS \$13.680,00
Product owner	36	ARS \$400,00	ARS \$14.400,00
Desarrollador Back-End 1	128	ARS \$360,00	ARS \$46.080,00
Desarrollador Back-End 2	128	ARS \$360,00	ARS \$46.080,00
Desarrollador Back-End 3	128	ARS \$360,00	ARS \$46.080,00
Desarrollador Front-End 1	128	ARS \$360,00	ARS \$46.080,00
Desarrollador Front-End 2	128	ARS \$360,00	ARS \$46.080,00
QA	32	ARS \$325,00	ARS \$10.400,00
Diseñador 1	36	ARS \$320,00	ARS \$11.520,00
Diseñador 2	36	ARS \$320,00	ARS \$11.520,00
Diseñador 3	36	ARS \$320,00	ARS \$11.520,00
<b>Total</b>			<b>ARS \$303.440,00</b>

Julio			
Nombre del recurso	Horas	\$ / hora	Sub total
Scrum Master	120	ARS \$380,00	ARS \$45.600,00
Product owner	128	ARS \$400,00	ARS \$51.200,00
Desarrollador Back-End 1	48	ARS \$360,00	ARS \$17.280,00
Desarrollador Back-End 2	48	ARS \$360,00	ARS \$17.280,00
Desarrollador Back-End 3	48	ARS \$360,00	ARS \$17.280,00
Desarrollador Front-End 1	168	ARS \$360,00	ARS \$60.480,00
Desarrollador Front-End 2	48	ARS \$360,00	ARS \$17.280,00
Analista 1	120	ARS \$300,00	ARS \$36.000,00
Diseñador 1	120	ARS \$320,00	ARS \$38.400,00
<b>Total</b>			<b>ARS \$300.800,00</b>

Agosto			
Nombre del recurso	Horas	\$ / hora	Sub total

Scrum Master	48	ARS \$380,00	ARS \$18.240,00
Product owner	48	ARS \$400,00	ARS \$19.200,00
Desarrollador Back-End 1	110	ARS \$360,00	ARS \$39.600,00
Desarrollador Back-End 2	110	ARS \$360,00	ARS \$39.600,00
Desarrollador Front-End 1	150	ARS \$360,00	ARS \$54.000,00
QA	208	ARS \$325,00	ARS \$67.600,00
DevOps	8	ARS \$330,00	ARS \$2.640,00
Analista 1	40	ARS \$300,00	ARS \$12.000,00
Diseñador 1	40	ARS \$320,00	ARS \$12.800,00
<b>Total</b>			<b>ARS \$265.680,00</b>

<b>Septiembre</b>			
<b>Nombre del recurso</b>	<b>Horas</b>	<b>\$ / hora</b>	<b>Sub total</b>
Product owner	8	ARS \$400,00	ARS \$3.200,00
Desarrollador Back-End 1	132	ARS \$360,00	ARS \$47.520,00
Desarrollador Back-End 2	132	ARS \$360,00	ARS \$47.520,00
Desarrollador Front-End 1	132	ARS \$360,00	ARS \$47.520,00
QA	528	ARS \$325,00	ARS \$171.600,00
<b>Total</b>			<b>ARS \$317.360,00</b>

<b>Octubre</b>			
<b>Nombre del recurso</b>	<b>Horas</b>	<b>\$ / hora</b>	<b>Sub total</b>
Product owner	48	ARS \$400,00	ARS \$19.200,00
Desarrollador Back-End 1	117	ARS \$360,00	ARS \$42.120,00
Desarrollador Back-End 2	61	ARS \$360,00	ARS \$21.960,00
Desarrollador Back-End 3	54	ARS \$360,00	ARS \$19.440,00
Desarrollador Front-End 1	61	ARS \$360,00	ARS \$21.960,00
Desarrollador Front-End 2	54	ARS \$360,00	ARS \$19.440,00
QA	240	ARS \$325,00	ARS \$78.000,00
DevOps	40	ARS \$330,00	ARS \$13.200,00

Analista 1	200	ARS \$300,00	ARS \$60.000,00
<b>Total</b>			<b>ARS \$295.320,00</b>

Noviembre			
Nombre del recurso	Horas	\$ / hora	Sub total
Analista 1	8	ARS \$300,00	ARS \$2.400,00
<b>Total</b>			<b>ARS \$2.400,00</b>

Salario Total Anual	
Mes	Sub Total
Marzo	ARS \$256.320,00
Abril	ARS \$94.480,00
Mayo	ARS \$235.680,00
Junio	ARS \$303.440,00
Julio	ARS \$300.800,00
Agosto	ARS \$265.680,00
Septiembre	ARS \$317.360,00
Octubre	ARS \$295.320,00
Noviembre	ARS \$2.400,00
<b>Total</b>	<b>ARS \$2.071.480,00</b>

Costo mensual (mantenimiento e innovación)			
Nombre recurso	Hora	\$ / hora	Sub total
Desarrollador Front-End	80	ARS 624	ARS 50,000.00
Desarrollador Back-End	80	ARS 624.00	ARS 50,000.00
QA	80	ARS 750.00	ARS 60,000.00
<b>Total</b>			<b>ARS 160,000.00</b>

### Costos de tecnologías



A continuación se detallarán los costos por servicios e infraestructura, según los valores que maneja el proveedor en su plataforma.

En este caso, el proveedor escogido es *Google Cloud* ([cloud.google.com](https://cloud.google.com)).

En cuanto se refiere a guardar las imágenes de docker en el repositorio de imágenes de google cloud, los precios son los siguientes:

Usage	Price (per GB / month)
Up to 0.5 GB	Free
Over 0.5 GB	\$0.10

The pricing is the same for both regional and multi-regional repositories.

Figura 230: Costo de guardar imágenes en el repositorio de Google Cloud

En cuanto a los costos de los clusters de kubernetes, Google Cloud sigue una política particular:

La tarifa de administración de clústeres de \$ 0.10 (15 pesos) por clúster por hora (cobrada en incrementos de 1 segundo) se aplica a todos los clústeres de GKE (Google Kubernetes Engine) independientemente del modo de operación, el tamaño del clúster o la topología.

El nivel gratuito de GKE proporciona \$74.40 (11544 pesos) en créditos mensuales por cuenta de facturación. Si solo usa un único clúster de piloto automático o zona, este crédito cubrirá al menos el costo total de ese clúster cada mes. Los créditos de nivel gratuito no utilizados no se transfieren y no se pueden aplicar a ningún otro SKU.

Organization's billing accounts	Autopilot cluster hours per month	Regional cluster hours per month	Zonal cluster hours per month	Free tier credit used	Total monthly GKE cluster management fee (at \$0.10/hour per cluster)
account_1	744	0	0	\$74.40	\$0
account_2	0	1000	500	\$50	\$100
account_3	1000	1000	1000	\$74.40	\$225.60

Figura 231: Costo de utilización de GKE (Google Kubernetes Engine)

Las bases de datos estarán ejecutándose bajo el plan gratuito de Mongo Atlas, por lo cual no se genera ningún costo adicional.

### 3.5 Análisis de riesgos

El objetivo del análisis de riesgos es cuantificar los riesgos a los que está sujeto la evolución del proyecto, la pérdida esperada y el costo de tomar medidas de control.

Cuando se estima que el costo es menor que la pérdida que uno espera, se debe realizar la acción programada.

Para realizar el cálculo del riesgo se tiene en cuenta la probabilidad de que dicho riesgo ocurra y los posibles impactos de los mismos:

La probabilidad e impacto se pueden clasificar en *Bajo, Medio y Alto*. (1, 2 y 3 respectivamente).

Esta clasificación se efectúa para determinar cuáles riesgos son más críticos y merecen que tengan más atención focalizada.

#### Definición de Riesgos

A continuación se detallan los riesgos que pueden afectarnos en las etapas de avance del proyecto, junto a qué tan probable son de ocurrir y su impacto:

#	Descripción	Probabilidad de ocurrencia	Impacto	Riesgo (Probabilidad x Impacto)
1	Requerimientos mal definidos	3	3	9
2	Definición Incorrecta de alcances	1	3	3
3	Pérdida de información por mala documentación	1	3	3
4	Mal clima laboral	2	3	6
5	Cambios de requerimientos	1	3	3
6	Errores en el diseño	2	3	6
7	Errores en la selección de tecnologías	1	2	2

8	Incorrecta estimación de tiempos	2	3	6
9	Incorrecta estimación de costos	1	1	1
10	Errores en el código	2	2	4
11	Errores en la implementación	2	2	4
12	Falta de comunicación	3	2	6

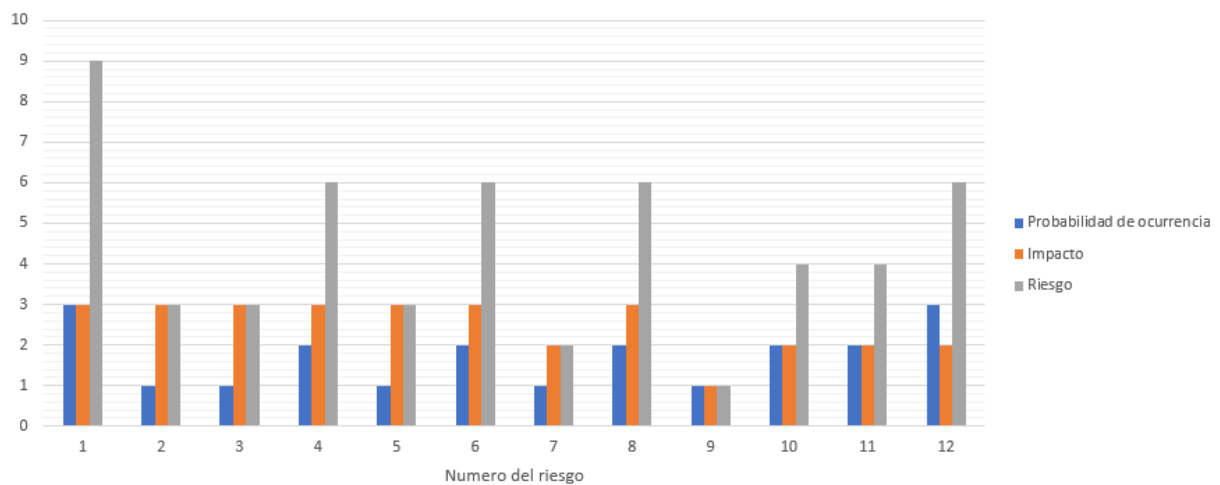


Figura 232: Gráfica del análisis de los riesgos

En el apartado siguiente realizaremos una clasificación de los riesgos para luego representarlos en una *Matriz de Riesgos*.

Los riesgos que tengan un puntaje de 6 - 9 serán **críticos**.

Los riesgos que tengan un puntaje de 3 - 4 serán **moderados**.

Los riesgos que tengan un puntaje de 1 - 2 serán **bajos**.

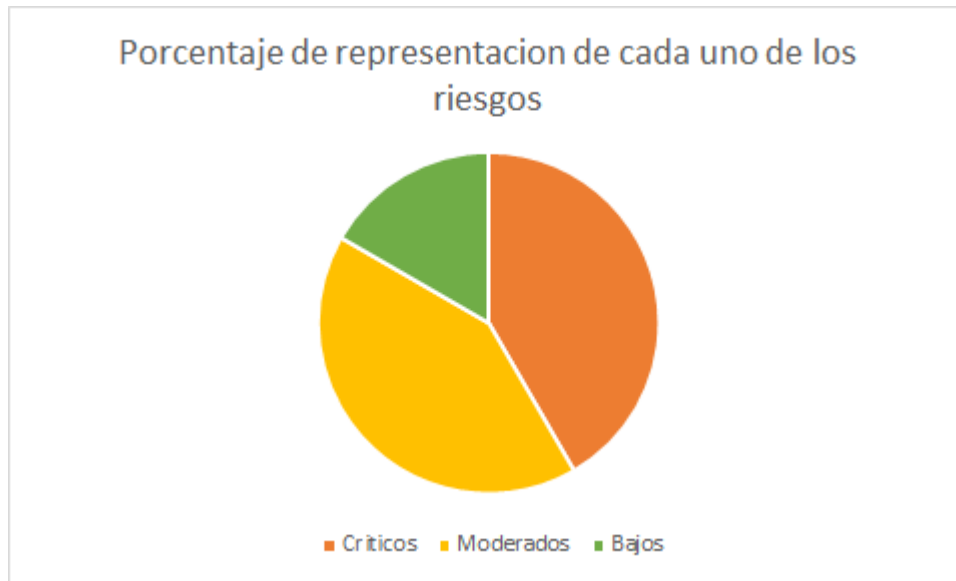


Figura 233: Gráfica que muestra el porcentaje de cada un de los riesgos

### Matriz de Riesgos

		Probabilidad de ocurrencia		
		Bajo	Medio	Alto
Impacto	Alto	Definición Incorrecta de alcance, Cambios de requerimientos, Pérdida de Información por mala documentación	Errores en el diseño, Mal clima laboral de los integrantes, Mala estimación de tiempos	Mala definición de requerimientos
	Medio	Mala elección de la tecnología	Errores en el código, Errores en la implementación	Falta de comunicación
	Bajo	Mala estimación de costos		

Descripción del Riesgo	Medida preventiva
------------------------	-------------------

Requerimientos mal definidos	<p>Estar en constante contacto con el cliente. Realizar un análisis más profundo sobre las necesidades de los clientes. De forma que se conozca completamente el proceso que se quiere implementar. Expresar claramente los requisitos y asegurarse que todos entiendan los objetivos. Elaborar el relevamiento con el mayor detalle posible.</p>
Definición Incorrecta de alcances	<p>Estar en constante contacto con el cliente. Volver a definir los alcances del proyecto y los tiempos junto con el cliente para que se adecue a la situación</p>
Mal clima laboral	<p>Hacer actividades grupales y juegos para aliviar la tensión entre los miembros del grupo.</p>
Cambios de requerimientos	<p>Tener un buen control de las actividades críticas, además de mantener reuniones frecuentes con el cliente, mostrar avances y mantener una buena comunicación.</p>
Errores en la selección de tecnologías	<p>Ejecutar un análisis y estudio previo de las tecnologías a utilizar. Efectuar una capacitación continua en dicha tecnología.</p>
Falta de comunicación	<p>Hacer actividades donde participen los miembros del grupo para mejorar la comunicación.</p>
Incorrecta estimación de tiempos.	<p>Determinar los tiempos de las tareas con suficiente holgura para atender desviaciones. Asignar tareas a los miembros más idóneos para llevarlas a cabo de forma exitosa. Identificar los caminos críticos para evitar reasignaciones.</p>

### 3.6 Análisis de impacto ambiental

Para realizar el impacto ambiental vamos a tener en cuenta las siguientes variables:

**Signo:** Si produce una mejora o un daño en el ambiente

Positivo (+)

Negativo (-)

**Magnitud:** Representa que tan fuerte es el impacto

Alta (3)

Media (2)

Baja (1)

**Alcance:** Donde repercute el cambio

Puntual (1)

Local (2)

Global (3)

**Persistencia:** Duración del efecto

Fugaz (1)

Temporal (2)

Permanente (3)

## Consumo de internet

El uso de internet genera contaminación ambiental, y al formar parte de ella la tenemos que tener en cuenta. La huella ecológica del tráfico digital equivale a un consumo aproximado del 7% de la electricidad mundial, según el informe de Greenpeace. Cada búsqueda que realizamos en internet libera al medio ambiente 0.2 gramos de CO<sub>2</sub>, lo que, según el propio Google, significa que el efecto invernadero provocado por mil búsquedas equivaldría a conducir un coche durante un kilómetro. El streaming de vídeos también está en pleno auge como demandante de datos. Ya en 2015, representaba el 63% del tráfico global de internet.

## Consumo de papel

El uso del sistema permitirá reducir considerablemente el uso de papel para la gestión de los eventos en Mendoza, ya que actualmente todo se hace a mano. Además de evitar el uso de papel también se evita el uso de titán que es otro gran contaminante para el medio ambiente.

## Consumo de energía

Al tratarse de un sistema que está completamente en la nube de nuestro lado solamente vamos a tener consumo de energía con las computadoras prendidas mientras trabajamos en el proyecto.

Por parte de Google va a requerir de servidores capaces de almacenar la plataforma para que pueda ser consumida por los interesados desde cualquier parte del mundo.

La cantidad de energía que consumen los servidores de datos se ha estimado en 205 teravatios por hora de electricidad (TWh), o el equivalente al 1% de la que consume el mundo en un año.

Debido a estos grandes requisitos de electricidad para enfriar los centros de datos, lugares con climas fríos, que cuenten con abundantes y confiables fuentes de alimentación eléctrica, son los más atractivos.

### Difusión de eventos

Al realizar la aplicación la difusión de eventos deportivos en la provincia se va a ver incrementado, ya que anteriormente no existía ningún tipo de aplicación similar para la zona.

### Acceso a eventos

El uso del sistema le permitirá a los distintos usuarios poder conocer los eventos próximos cercanos a su ciudad y la posibilidad de inscribirse a los mismos, facilitando enormemente la tarea de inscripción, ya que actualmente todo se realiza a mano.

### Centralización

El uso de la aplicación permitirá tener todos los eventos deportivos en un solo sitio.

### Adaptabilidad

Fácil adaptación por parte de los usuarios a la aplicación, debido a su interfaz fácil e intuitiva de usar.

Impacto		Signo	Magnitud	Alcance	Persistencia	Resultado
Social	Difusión de eventos	+	2	1	2	4
	Acceso a eventos	+	3	1	2	6
	Centralización	+	2	2	2	8
Energético	Consumo de internet	-	3	2	1	-6
	Consumo de energía	-	3	1	1	-3

Ambiental	Consumo de papel	+	2	3	3	18
Usuarios	Adaptabilidad	+	1	2	2	4
<b>Impacto ambiental</b>						31

Para evaluar si el proyecto es viable desde el punto de vista ambiental o no se multiplica todos los valores de las características (Signo, Magnitud, Alcance, Persistencia) y luego multiplicarlo por su signo, en caso de ser negativo se cuenta como un -1 y positivo un 1.

#### Conclusión del análisis de impacto ambiental

Se concluye que el análisis ambiental resulta positivo, lo que implicaría que no hay ningún tipo de trabas para el desarrollo e implementación del sistema.

#### Conclusión del análisis.

El proyecto se adapta al estudio de factibilidad en cada uno de los puntos descritos anteriormente con facilidad, al no poseer una gran cantidad de riesgos críticos que modifiquen el curso del proyecto de manera drástica. Además, al ser un producto de software, el análisis de impacto ambiental resulta increíblemente favorable al no poseer prácticamente consumo de internet, papel y energía en comparación a la mayoría de las industrias.





# Desarrollo e Implementación

En esta sección se describe la parte más significativa del código a utilizar. También se presentará la arquitectura y las tecnologías empleadas para su desarrollo. Finalmente, se explican los pasos requeridos para su implementación y configuración inicial, incluyendo el manual de usuario y el guionado de los videos del mismo.

## Programación y documentación

### Tecnologías utilizadas

El stack tecnológico de gestión de la configuración está compuesto por:

Gitlab



Figura 234: Logo de GitLab

GitLab es la plataforma DevOps, entregada como una sola aplicación. Esto hace que GitLab sea único y crea un flujo de trabajo de software optimizado, liberando a su organización de las limitaciones de una cadena de herramientas ensamblada. Descubra cómo GitLab ofrece una visibilidad incomparable y niveles más altos de eficiencia en una sola aplicación a lo largo del ciclo de vida de DevOps.

En nuestro proyecto usamos GitLab para administrar el versionado de la aplicación e implantar integración continua y entrega continua agilizando el desarrollo del mismo.

El stack tecnológico de pruebas está compuesto por:

Jmeter:



Figura 235: Logo de JMeter

JMeter es un proyecto de Apache que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el rendimiento de una variedad de servicios, con énfasis en aplicaciones web.

JMeter puede ser usado como una herramienta de pruebas unitarias para conexiones de bases de datos con JDBC, FTP, LDAP, Servicios web, JMS, HTTP y conexiones TCP genéricas. JMeter puede también ser configurado como un monitor, aunque es comúnmente considerado una solución ad-hoc respecto de soluciones avanzadas de monitoreo.

En nuestro proyecto utilizamos jmeter para realizar las pruebas funcionales, de seguridad y principalmente como herramienta de generación diseño y ejecución de pruebas de carga y estrés, que es donde esta herramienta más se destaca.

Jasmine



Figura 236: Logo de Jasmine

Jasmine es un marco de desarrollo impulsado por el comportamiento para probar código JavaScript. No depende de ningún otro marco de JavaScript. No requiere DOM. Y tiene una sintaxis clara y obvia para que pueda escribir pruebas fácilmente.

El desarrollo dividido en dos partes, front-end, la parte del sistema que interactúa directamente con el usuario, y back-end, la parte que gestiona los datos, la seguridad y la lógica de negocio.

El stack tecnológico de frontend está compuesto por:

Angular



Figura 237: Logo Framework Angular

Angular es un framework open source desarrollado por Google para facilitar la creación y programación de aplicaciones web de una sola página, las webs SPA (Single Page Application).

Angular separa completamente el frontend y el backend en la aplicación, evita escribir código repetitivo y mantiene todo más ordenado gracias a su patrón MVC (Modelo-Vista-Controlador) asegurando los desarrollos con rapidez, a la vez que posibilita modificaciones y actualizaciones.

En una web SPA aunque la velocidad de carga puede resultar un poco lenta la primera vez que se abre, navegar después es totalmente instantáneo, ya que se ha cargado toda la página de golpe.

Solamente es una ruta la que se tiene que enviar el servidor, y Angular lo que hace 'por debajo' es cambiar la vista al navegar para que dé la apariencia de una web normal, pero de forma más dinámica.

Entre otras ventajas, este framework es modular y escalable adaptándose a nuestras necesidades y al estar basado en el estándar de componentes web, y con un conjunto de interfaz de programación de aplicaciones (API) permite crear nuevas etiquetas HTML personalizadas que pueden reutilizarse.

Redux

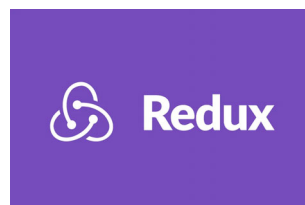


Figura 238: Logo de Redux

Redux es una librería JavaScript que emite actualizaciones de estado en respuesta a acciones, con la peculiaridad de realizar dichas modificaciones a través de objetos sencillos, que reciben el nombre de acciones, y no a través de cambios directos en el estado. Redux es un contenedor predecible del estado de aplicaciones JavaScript.

Te ayuda a escribir aplicaciones que se comportan de manera consistente, corren en distintos ambientes (cliente, servidor y nativo), y son fáciles de probar. Además de eso, provee una gran experiencia de desarrollo, gracias a edición en vivo combinado con un depurador sobre una línea de tiempo.

Puedes usar Redux combinado con React, o cualquier otra librería de vistas. Es muy pequeño y no tiene dependencias.

## Bootstrap



Figura 239: Logo Bootstrap

Bootstrap es un framework que combina CSS y JavaScript para estilizar los elementos de una página HTML. Permite mucho más que, simplemente, cambiar el color de los botones y los enlaces.

Esta es una herramienta que proporciona interactividad en la página, por lo que ofrece una serie de componentes que facilitan la comunicación con el usuario, como menús de navegación, controles de página, barras de progreso y más.

Además de todas las características que ofrece el framework, su principal objetivo es permitir la construcción de sitios web responsive para dispositivos móviles.

Esto significa que las páginas están diseñadas para funcionar en desktop, tablets y smartphones, de una manera muy simple y organizada.

El stack tecnológico de backend está compuesto por:

## Node.js



Figura 240: Logo de nodeJS

En la parte de los servicios del sistema utilizaremos Node.js® que es una tecnología ideada como un entorno de ejecución de JavaScript orientado a eventos asíncronos, diseñada fundamentalmente para crear aplicaciones network escalables.

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, servidores web.

Express.js



Figura 241: Logo de express

Express.js es un framework para Node.js, es un software gratuito y de código abierto bajo la licencia MIT. Está diseñado para crear aplicaciones web y API. Se le ha llamado el marco de servidor estándar de facto para Node.js.

Es un framework inspirado en Sinatra, lo que significa que es relativamente mínimo con muchas características disponibles como complementos. Express es el componente de back-end de desarrollo populares como MEAN, MERN o MEVN, junto con el software de base de datos MongoDB y un marco o biblioteca de front-end de JavaScript.

Docker

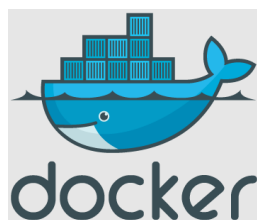


Figura 242: Logo de docker

Docker es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.

Docker utiliza características de aislamiento de recursos del kernel Linux, tales como cgroups y espacios de nombres (namespaces) para permitir que "contenedores"

independientes se ejecuten dentro de una sola instancia de Linux, evitando la sobrecarga de iniciar y mantener máquinas virtuales.

El soporte del kernel Linux para los espacios de nombres aísla la vista que tiene una aplicación de su entorno operativo, incluyendo árboles de proceso, red, ID de usuario y sistemas de archivos montados, mientras que los cgroups del kernel proporcionan aislamiento de recursos, incluyendo la CPU, la memoria, el bloque de E/S y de la red. Desde la versión 0.9, Docker incluye la biblioteca libcontainer como su propia manera de utilizar directamente las facilidades de virtualización que ofrece el kernel Linux, además de utilizar las interfaces abstraídas de virtualización mediante libvirt, LXC (Linux Containers) y systemd-nspawn.

## Kubernetes



Figura 243: Logo de kubernetes

Kubernetes es una plataforma open source para administrar cargas de trabajo y servicios, facilitando la automatización y eliminando muchos de los procesos manuales durante la implementación y escalabilidad de las aplicaciones en contenedores. En este sentido, nos ofrece una mayor simpleza en plataformas como servicio (PaaS) y una mayor flexibilidad en la infraestructura como servicio (IaaS), permitiendo la portabilidad entre proveedores de infraestructura. Se puede desplegar en diversos entornos cloud y soporta múltiples runtimes de contenedores (docker, rkt, cri-o o frakti).

Aunque en el origen de esta interesante plataforma portable y extensible fue Google el encargado del desarrollo y diseño, en 2014 el gigante tecnológico decidió liberar el proyecto y donarlo a la Cloud Native Computing Foundation (Linux Foundation). Al igual que Docker, Kubernetes se ha convertido en uno de los estándares como gestor de contenedores de software dentro de la industria.

## MongoDB



Figura 244: Logo de mongoDB

En lo que respecta a la base de datos utilizaremos MongoDB, el cual es un motor de base de datos no relacional que posee muchas ventajas porque al ser un esquema documental, la información que se almacena no se guarda bajo un formato rígido de almacenamiento.

En lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos BSON (una especificación similar a JSON) con un esquema dinámico, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

MongoDB es una base de datos adecuada para su uso en producción y con múltiples funcionalidades.

Esta base de datos se utiliza mucho en la industria contando con implantaciones en empresas como MTV Network, Craigslist, Foursquare, etc.

El código fuente está disponible para los sistemas operativos Windows, GNU/Linux, OS X y Solaris.

Redis



Figura 245: Logo de redis

Redis, que significa Remote Dictionary Server (servidor de diccionarios remoto), es un rápido almacén de datos clave-valor en memoria de código abierto que se puede utilizar como base de datos, caché, agente de mensajes y cola. Redis ofrece ahora tiempos de respuesta inferiores al milisegundo, lo que permite que se realicen millones de solicitudes por segundo para aplicaciones en tiempo real de videojuegos, tecnología publicitaria, servicios financieros, sanidad e IoT.

Redis es una opción muy habitual en aplicaciones de almacenamiento en caché, administración de sesiones, videojuegos, tablas de clasificación, análisis en tiempo real, datos geoespaciales, servicios de vehículos compartidos, chat/mensajería, streaming de contenido multimedia y publicación/suscripción.



Mongoose



Figura 246: Logo de mongoose

Mongoose es una herramienta de modelado de objetos de MongoDB diseñada para funcionar en un entorno asincrónico. Mongoose admite promesas y callbacks.

Cloudinary



Figura 247: Logo de Cloudinary

Cloudinary es una empresa de tecnología SaaS con sede en Santa Clara, California, con oficinas en Israel, Inglaterra, Polonia y Singapur. La empresa ofrece servicios de gestión de imágenes y vídeos basados en la nube.

vs code



Figura 248: Logo de VS Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código. También es personalizable, por lo que los usuarios pueden cambiar el tema del editor, los atajos de teclado y las preferencias. Es gratuito y de código abierto, aunque la descarga oficial está bajo software privativo e incluye características personalizadas por Microsoft.

## WebStorm



Figura 249: Logo de WebStorm

WebStorm es un entorno de desarrollo integrado para JavaScript y las tecnologías relacionadas. Al igual que otros IDE de JetBrains, hace que su experiencia de desarrollo sea más agradable, automatiza las tareas repetitivas y le ayuda a gestionar las tareas complejas con facilidad.

## Estructura del proyecto en Visual Code

### Frontend

El front-end consiste en una aplicación web, desarrollada con el framework Angular y utilizando el lenguaje TypeScript, que luego se traduce a JavaScript. Está estructurado en base a la organización que ya provee Angular pero con algunas modificaciones para poder trabajar mejor con componentes. En la carpeta assets se encuentran las imágenes, archivos js de terceros y un módulo de iconos para importar diferentes iconos de la librería fontawesome. En la carpeta environments (Carpeta creada automáticamente) se configura la url base con la que se apuntará a los endpoints para que en caso de ser necesario no cambiarla desde todos los servicios sino desde un solo lugar. En la carpeta node\_modules contiene todos los archivos de terceros necesarios para el funcionamiento de la aplicación. En la carpeta app hay varias carpetas que a su vez pueden contener subcarpetas. A continuación se describirán el contenido de las carpetas más importantes:

- **component:** Contendrá componentes hijos que podrán utilizar los componentes padres que se encuentran en la carpeta pages. Se puede llegar a utilizar un componente hijo en varios componentes padres pero por lo general para estos casos se usará la carpeta shared.
- **guards:** Contendrá los guards que se usarán para validar que los usuarios que quieren entrar a una página tengan los permisos adecuados.

- **helpers:** Contendrá archivos de typescript con funciones genéricas que se usan en varios componentes.
- **Modals:** En modals, se diseñan los cuadros modales con sus estilos y funciones para las diferentes pantallas y fines del proyecto.
- **Materials:** Configuraciones e importaciones necesarios para la utilización de los componentes de la librería “Materials”
- **models:** En esta carpeta nos encontramos con los modelos de las diferentes entidades de la aplicación.
- **pages:** En esta carpeta nos encontramos con los componentes que requieren estar autenticado para ser vistos.
- **services:** Nos encontramos con los services de la aplicación que por lo general se usarán para comunicarse con la base de datos o realizar algunas funciones globales.
- **shared:** En esta carpeta nos encontramos con todos los módulos que se usan en casi todos los componentes de la aplicación. Un ejemplo claro de esto es el componente “navbar”.
- **Assets:** Contiene las imágenes e iconos necesarios para mostrar en el sistema.
- **store:** En esta carpeta nos encontramos con todos los archivos necesarios para poder implementar redux en la aplicación.
- **Pipes:** Contiene los pipes necesarios para la correcta transformación de datos en la aplicación.
- **componentes:** Los componentes que no necesiten seguridad como la landing page, el login, el signup entre otros se encontrarán en el directorio base de app.
- **Environments:** Contiene las variables de entorno para configurar la conexión a los distintos ambientes
- **Páginas generales:** Las páginas generales son aquellas que no pertenecen a un módulo en particular y no pueden ser incluidas dentro de los módulos dentro de la carpeta “pages” . Estas son:
  - **Páginas-de-inicio**
  - **page-not-found**
  - **login**
  - **landing-page**
  - **signup**

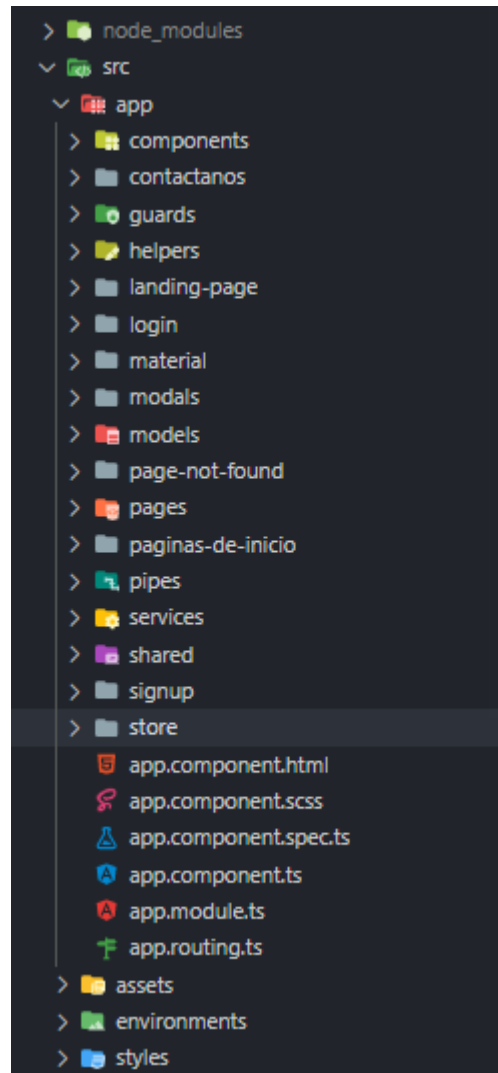


Figura 250: Estructura del proyecto FrontEnd

## Backend

Para el desarrollo del backend se va a utilizar una interfaz REST para conectar varios sistemas basados en el protocolo HTTP.

El desarrollo con NodeJS de un API REST es especialmente indicado, dado que nos permite construir el backend con el mismo lenguaje con el que se construye el front-end, evitando la fatiga de pasar de un lenguaje a otro cuando se desarrollan ambas capas de un proyecto web. Además, dadas las características de esta plataforma, también se hace muy adecuada para el desarrollo RESTful dado que Node, al no ser bloqueante, puede atender a más usuarios de manera concurrente.

En NodeJS, vamos a utilizar Express que es uno de los frameworks más potentes para el desarrollo de APIs y aplicaciones en general.

Para manipular los datos de la base de datos vamos a usar mongoose que es un Object Document Mapper (ODM). Esto significa que Mongoose le permite definir objetos con un

esquema fuertemente tipado que se asigna a un documento MongoDB. Mongoose proporciona una increíble cantidad de funcionalidades para crear y trabajar con esquemas. En cuanto al código del backend, este se organiza bajo la siguiente estructura:

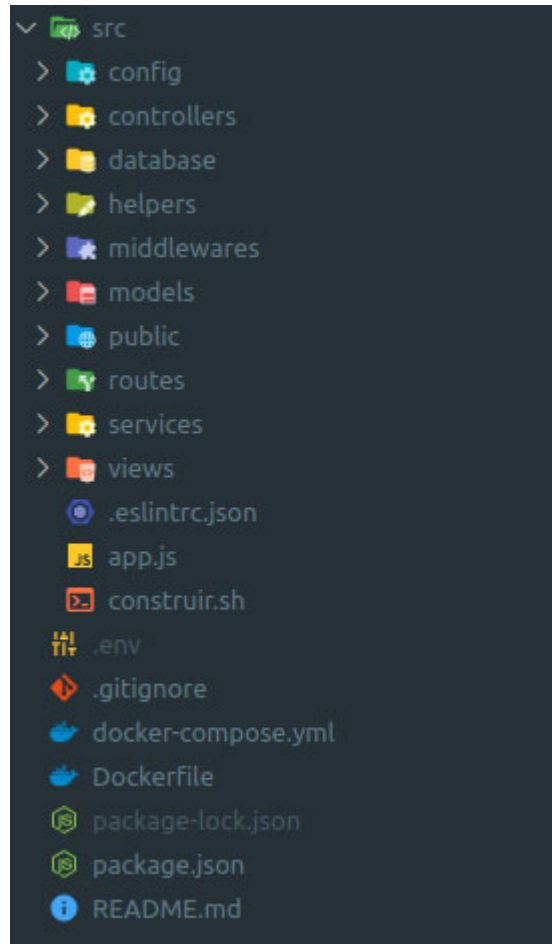


Figura 251: Estructura del proyecto BackEnd

Para explicar la estructura del proyecto, vamos a explicar cada una de las áreas:

- **Controllers:** Métodos que van a encargarse de manipular los datos y entregarlos a los usuarios que lo requieren.
- **Database:** Métodos que se encargan de conectarse con la base de datos
- **Helpers:** Métodos que se encargan de funciones auxiliares que se reutilizan en distintas partes del sistema
- **Middlewares:** Métodos que ofrecen servicios y funciones comunes para las aplicaciones
- **Models:** Schemas que definen la estructura de la base de datos
- **Routes:** Métodos que reciben peticiones del exterior y se definen las rutas que van a poder ser accedidas desde afuera.
- **Services:** Métodos que actúan como intermediarios entre los controllers y las llamadas a la base de datos.

## Estilo de codificación

- Los lineamientos de estilo utilizados en el proyecto derivan de la guía de estilo para Angular.
- Vamos a utilizar CamelCase que es un estilo de escritura que se aplica a frases o palabras compuestas, específicamente se utilizará lowerCamelCase, cuando la primera letra de cada una de las palabras es mayúscula a excepción del inicio de la clase. Ejemplo: CategoriaEvento.
- Si una variable no va a cambiar de valor durante todo el ciclo de vida usar const.
- Para propiedades y métodos, también vamos a utilizar la nomenclatura lowerCamelCase
- Evitar utilizar prefijos como ‘\_’ delante del nombre de un atributo o función.
- No incluir una ruta absoluta en el código. Se recomienda conseguirla mediante llamadas a funciones, ficheros de configuración o usar rutas relativas.
- Se recomienda hacer uso del paso de parámetros a función con valores por defecto ya que hará que tengamos el código más limpio.
- Para igualar dos valores vamos a utilizar la siguiente expresión ‘variable1 == variable2’
- Las declaraciones no deben estar en una única línea ya que son más difíciles de leer.

## Organización para el desarrollo

Para llevar a cabo el desarrollo del proyecto, nos dividimos de la siguiente manera:

- Se preparó el entorno de GIT para gestionar el código del proyecto.
  - Se creó un grupo de trabajo llamado Sportinev en Gitlab para almacenar el repositorio que se llama front-end, el cual contiene el código del mismo.
  - Se creó el repositorio llamado Sportinev-Angular
  - Dentro del repositorio hay una rama principal llamada master, la cual contiene la última versión de la aplicación
  - Se creó una rama llamada develop, donde se desarrollará y se probarán las nuevas funcionalidades y se las integrará a las otras
  - Por cada nueva funcionalidad se crea una rama con el formato “feature-nombre-funcionalidad” y se desarrolla la funcionalidad en esa misma rama.
  - Luego de terminar el desarrollo de la funcionalidad se mergea la rama recién mencionada con la rama develop.
  - Una vez que se decida aumentar nuestra versión productiva, se mergea la rama develop en la rama master y se despliega la aplicación en el ambiente productivo.
- Creamos una cuenta en Mongo Atlas
- Se creó un cluster de bases de datos en Mongo Atlas
- Se configuraron las credenciales para el cluster

- Se creó la bases de datos en Cluster de Mongo Atlas
- Una vez creada la base de datos generamos el string de conexión que irá en nuestro archivo de configuración del entorno en el backend (.env)
- Se creó un cluster de kubernetes en google usando Google Kubernetes Engine
- Creamos el cluster en una zona geográfica adecuada para tener una buena latencia
- Luego de crear el cluster creamos los permisos para los integrantes del proyecto
- Una vez creado el cluster y creados los nodos dentro del cluster, eligiendo nodos de dimensiones adecuadas para nuestro proyecto
- Luego de eso creamos los archivos de despliegue (.yaml) y los scripts para desplegar y armar las imágenes (build.sh y deploy.sh)
- Además armamos un docker-compose para levantar las aplicaciones como contenedor en local y podes hacer pruebas localmente
- Una vez armada la versión estable productiva esta misma se desplegará en el cluster de kubernetes con el script de despliegue

Para desarrollar la organización fue:

- A quienes no habían programado nunca en Angular o NodeJS se les asignaron tareas fáciles y poco rebuscadas como ABMs para que fueran tomando confianza con la tecnología.
- Una vez familiarizados con el entorno de programación y las tecnologías, el equipo desarrollará mucho más rápido
- La parte del equipo que tiene más experiencia desarrollando, comenzó con las HU con mayor complejidad y prioridad y más visuales para el usuario final, de modo que, la calidad de estas historias y el tiempo de desarrollo fuera el más eficiente.
- Las Historias de Usuario poseen una escala cualitativa de prioridad ordenada con letras siendo A la más alta y descendiendo (A, B, C, etc.).

## Documentación módulo de seguridad

### HU N°1 SignUp Deportista

Planilla de documentación

<b>Nombre de la HU</b>	Sign up deportista
<b>Módulo seleccionado</b>	Seguridad
<b>Descripción breve de la HU</b>	Como potencial usuario con rol deportista quiero crear una cuenta para poder identificarme en el sistema.

<b>Actor</b>	Usuario
<b>Prioridad</b>	A
<b>Programador BackEnd</b>	Ignacio Starkloff
<b>Programador FrontEnd</b>	Juan Riera
<b>Tester</b>	Elvio Aruta
<b>Aprobado por</b>	Miguel Del Corso
<b>Denominación del programa</b>	<b>backend</b> “signup.controller.js” “signup.service.js” “signup.helper.js” <b>frontend</b> “SignUp.component.ts” “SignUp.component.html”
<b>Fecha inicio Programación</b>	09/08/2021
<b>Fecha finalización de programación</b>	13/08/2021
<b>Base de datos</b>	MongoDB Redis
<b>Colecciones involucradas</b>	Usuario Estado Usuario Estado Deportistas Rol UsuarioRol DeportistaTipoDeporte
<b>Casos de pruebas utilizados con Resultados de la prueba</b>	
TCS-1 TCI-5 Crear usuario deportista con el campo nombre con un array vacío=> PASS Crear usuario deportista con el campo nombre con caracteres especiales=> PASS Crear usuario deportista con el campo nombre con un entero => PASS Crear usuario deportista con el campo nombre con un número con coma => PASS Crear usuario deportista con el campo nombre con un número con punto => PASS Crear usuario deportista con el campo nombre con un espacio en blanco=> PASS Crear usuario deportista con el campo nombre con un json mal formulado => PASS Crear usuario deportista con el campo nombre con los caracteres: \' => PASS Crear usuario deportista con el campo nombre con dos caracteres: \ => PASS Crear usuario deportista con el campo apellido con un array vacío=> PASS Crear usuario deportista con el campo apellido con caracteres especiales=> PASS Crear usuario deportista con el campo apellido con un entero => PASS Crear usuario deportista con el campo apellido con un número con coma => PASS	



Crear usuario deportista con el campo apellido con un número con punto => PASS  
 Crear usuario deportista con el campo apellido con un espacio en blanco=> PASS  
 Crear usuario deportista con el campo apellido con un json mal formulado => PASS  
 Crear usuario deportista con el campo apellido con los caracteres: \" => PASS  
 Crear usuario deportista con el campo apellido con dos caracteres: \ => PASS  
 Crear usuario deportista con el campo correo con un solo mail vacío=> PASS  
 Crear usuario deportista con el campo correo con un mail sin caracteres previos al arroba => PASS  
 Crear usuario deportista con el campo correo con el carácter: " => PASS  
 Crear usuario deportista con el campo correo con un mail con más de un punto luego de el arroba => PASS  
 Crear usuario deportista con el campo correo con un mail terminando en punto => PASS  
 Crear usuario deportista con el campo correo con un mail sin puntos luego de el arroba => PASS  
 Crear usuario deportista con el campo correo con un mail con caracteres especiales=> PASS  
 Crear usuario deportista con el campo correo con un mail con más de un @=> PASS  
 Crear usuario deportista con el campo correo con un mail con el carácter: -=> PASS  
 Crear usuario deportista con el campo correo con un mail con el carácter: \_=> PASS  
 Crear usuario deportista con el campo correo con un mail con menos de 2 caracteres luego del punto final=> PASS  
 Crear usuario deportista con el campo correo con un mail con menos de 1 caracter luego del arroba y previo al punto=> PASS  
 Crear usuario deportista con el campo correo con un mail con más de 100 caracteres=> PASS  
 Crear usuario deportista con el campo password con un array vacío=> PASS  
 Crear usuario deportista con el campo password con caracteres especiales=> PASS  
 Crear usuario deportista con el campo password con un entero => PASS  
 Crear usuario deportista con el campo password con un número con coma => PASS  
 Crear usuario deportista con el campo password con un número con punto => PASS  
 Crear usuario deportista con el campo password con un espacio en blanco=> PASS  
 Crear usuario deportista con el campo password con un json mal formulado => PASS  
 Crear usuario deportista con el campo password con los caracteres: \" => PASS  
 Crear usuario deportista con el campo password con dos caracteres: \ => PASS  
 Crear usuario deportista con el campo password con un json mal formulado => PASS  
 Crear usuario deportista con el campo rol con el campo vacío=> PASS  
 Crear usuario deportista con el campo rol con un rol inexistente=> PASS  
 Crear usuario deportista con el campo rol con un rol no habilitado=> PASS  
 Crear usuario deportista con el campo rol con un json mal formulado => PASS

**Historias de usuarios relacionadas**

HU 6 - Login  
HU 2 - SignUp club

Documentación del BackEnd

**URL** \${URL}/api/auth/signup

**Method** POST

## Body

De acuerdo con el rol deseado del usuario, los datos solicitados varían.

### Deportista

```
{
  "nombre": "Deportista",
  "apellido": "Deportista",
  "correo": "deportista@gmail.com",
  "password": "123456",
  "rol": "DEPORTISTA"
}
```

### Club

```
{
  "nombre": "Club",
  "CUIT": "30872138201",
  "teléfono": "2616510467",
  "correo": "club@gmail.com",
  "password": "123456",
  "localidad": "idLocalidad",
  "domicilio": "domicilio club",
  "latitud": 2312312312.12,
  "longitud": 2312312312.12,
  "rol": "CLUB"
}
```

### Árbitro

```
{
  "nombre": "Árbitro",
  "DNI": "41699065",
  "teléfono": "2616510467",
  "correo": "arbitro@gmail.com",
  "password": "123456",
  "provincia": "idProvincia",
  "deporte": "idDeporte",
  "rol": "ARBITRO"
}
```

### Organizador

```
{
  "nombre": "Organizador",
  "CUIT": "20416990655",
}
```

```
"telefono": "02616510467",  
"tipoOrganizador": "idTipoOrganizador",  
"correo": "organizador@gmail.com",  
"password": "123456",  
"provincia": "idProvincia",  
"rol": "ORGANIZADOR"  
}
```

### Patrocinador

```
{  
  "nombre": "Patrocinador",  
  "CUIT": "20416990655",  
  "telefono": "02616510467",  
  "correo": "patrocinador@gmail.com",  
  "password": "123456",  
  "rol": "PATROCINADOR"  
}
```

### Configuración de la ruta

```
router.post(  
  "/signup",  
  [  
    check("nombre", "El nombre es obligatorio").not().isEmpty(),  
    check("correo", "El correo no es válido").isEmail(),  
    check("password", "La contraseña debe tener más de 6 caracteres").isLength({  
      min: 6,  
    }),  
    check("rol").custom((rol) => rolValido(rol)),  
    validarCampos,  
  ],  
  asyncHandler(signup)  
);
```

Lo siguiente es ir al método signup ubicado en el authcontroller

```
const signup = async (req = request, res = response) => {  
  const body = req.body;  
  // Verificamos que no exista otro usuario con el correo ingresado  
  await emailUnico(body.correo);  
  // Verificamos que no exista un token válido para crear la cuenta  
  await tokenEmailUnico(body.correo);  
  // Verificamos si el rol con el que se está intentando crear la cuenta el usuario es  
  // válido  
  await verificarRolUsuario(body);  
  // Guardamos el token con toda la información ingresada
```

```
const token = await saveConfirmarCuentaService(body);  
// Enviamos un mail con el link para activar la cuenta  
await activarCuentaService(token, token.uid);  
  
res.json({  
  message:  
    "Se ha generado un link para activar su cuenta, por favor verifique su email y haga  
click en el link",  
});  
};
```

Nos encargamos verificar los datos acorde al rol seleccionado

```
const verificarRolUsuario = async (usuario) => {  
  const {  
    CUIT,  
    fechaNacimiento,  
    tipoOrganizador,  
    provincia,  
    apellido,  
    deporte,  
    departamento,  
    domicilio,  
    latitud,  
    longitud,  
    localidad,  
  } = usuario;  
  switch (usuario.rol) {  
    case "ORGANIZADOR":  
      if (!tipoOrganizador || !provincia || !CUIT) {  
        throw new BadRequest(  
          "El usuario debe tener un tipo de cuenta, estar asignado a una provincia y  
tener una CUIT que lo identifique para ser organizador"  
        );  
      } else {  
        await findTipoOrganizadorByIdService(tipoOrganizador);  
        await findProvinciaByIdService(provincia);  
      }  
      break;  
    case "DEPORTISTA":  
      if (!apellido, !fechaNacimiento, !departamento) {  
        throw new BadRequest(  
          "El usuario debe tener un apellido, una fecha de nacimiento y un departamento  
para ser deportista"  
        );  
      }  
      break;  
    case "ARBITRO":  
      if (!deporte) {  
        throw new BadRequest(  
          "El usuario debe tener un deporte para ser arbitro"  
        );  
      }  
    }  
  }  
};
```

```

    }
    await findDeporteByIdService(deporte);
    break;
  case "CLUB":
    if (!domicilio || !latitud || !longitud || !localidad) {
      throw new BadRequest(
        "El usuario debe tener un domicilio para ser club"
      );
    }
    await findLocalidadByIdService(localidad);
    break;
  case "PATROCINADOR":
    break;
  default:
    if (!CUIT) {
      throw new BadRequest(
        "El usuario debe tener una CUIT que lo identifique para ser patrocinador"
      );
    }
    throw new NotAcceptable(
      `El rol ${usuario.rol} no esta disponible para usted, por favor solicite al
      administrador si quiere tener este rol en el sistema`
    );
  }
};

```

Este método se encarga de verificar el email del usuario.

Cuando el usuario hace click sobre el link se activa el método confirmarCuenta en authcontroller.

```

const confirmarCuenta = async (req = request, res = response) => {
  try {
    const idToken = req.params.token;
    // Buscamos el token por su id el cual fue pasado por parámetro
    const token = await findTokenByIdService(idToken);
    // Si el token no existe mostrar mensaje que el link expiro
    if (!token) {
      res.render("respuesta", {
        layout: false,
        message: "El link ha expirado, por favor vuelva a generar uno nuevo",
      });
    }
    // Creamos el modelo del usuario
    const usuario = await crearModeloUsuario(token);
    // Eliminamos el token
    await deleteTokenService(idToken);
    // Guardamos al usuario
    await saveUsuarioService(usuario);

    res.render("respuesta", {

```

```
    layout: false,  
    message: "La cuenta se ha creado con éxito",  
  });  
} catch (error) {  
  console.log(error);  
  res.render("respuesta", {  
    layout: false,  
    message:  
      "Error, hubo un problema que no permite confirmar su cuenta, por favor  
comuníquese con el administrador",  
  });  
}  
};
```

Y procedemos a crear el modelo del deportista

```
const crearModeloDeportista = async (usuario) => {  
  // Datos que son necesarios para crear el modelo del deportista  
  const { DNI, apellido, fechaNacimiento, departamento, _id } = usuario;  
  if (apellido && fechaNacimiento && departamento && _id) {  
    // Buscamos el departamento en la base de datos  
    const modeloDepartamento = await findDepartamentoByIdService(departamento);  
    // Creamos modelo del deportista  
    const modeloDeportista = new Deportista({  
      apellido: apellido,  
      fechaNacimiento: fechaNacimiento,  
      departamento: modeloDepartamento,  
      DNI: DNI,  
      usuario: _id,  
    });  
    // Guardamos al deportista en la base de datos  
    await saveDeportistaService(modeloDeportista);  
    return modeloDeportista;  
  } else {  
    throw new BadRequest(  
      "El formato enviado no es el adecuado para crear un deportista"  
    );  
  }  
};
```

Pantallas

**SPORTINEV**

### Registrarse

---

Ingrese su nombre

Ingrese su teléfono o celular

Ingrese su email

Ingrese su contraseña 

Tipo de usuario 

**Registrarse**

Volver

Figura 252: Pantalla de registrarse

The image shows a registration form with a dark background and light text. The form contains the following fields and elements from top to bottom:

- Input field: "Ingrese su nombre"
- Input field: "Ingrese su teléfono o celular"
- Input field: "Ingrese su email"
- Input field: "Ingrese su contraseña" with an eye icon for visibility toggle.
- Dropdown menu: "Tipo de usuario" with "DEPORTISTA" selected.
- Input field: "Ingrese su apellido"
- Input field: "Fecha de nacimiento" with a calendar icon.
- Input field: "Ingrese su DNI"
- Dropdown menu: "País"
- Dropdown menu: "Provincia"
- Dropdown menu: "Departamento"
- Red button: "Registrarse"
- White button: "Volver"

Figura 253: Pantalla de registrarse como deportista



## Documentación del FrontEnd

### Componente padre SignUp.component.html

```

<div class="container mt-5">
  <!-- Logo de la aplicación -->
  <div class="row">
    <div class="col-12 text-center">
      
    </div>
  </div>
  <!-- Componente card nos da una estructura responsive solo hay que pasarle el
  form -->
  <app-card-form [componentePasado]=signupForm
  titulo="Registrarse"></app-card-form>
</div>

<!-- Form que se le enviara a card -->
<ng-template #signupForm>
  <form [formGroup]="form">
    <!-- Input para obtener nombre del nuevo usuario -->
    <div class="my-3">
      <mat-form-field appearance="legacy" style="width: 100%;">
        <mat-label>Ingrese su nombre</mat-label>
        <input matInput placeholder="Juan" formControlName="nombre"
autocomplete="off">
        <mat-error *ngIf="form.controls['nombre'].errors">El
          nombre es obligatorio</mat-error>
      </mat-form-field>
    </div>

    <!-- Input para obtener teléfono del nuevo usuario -->
    <div class="my-3">
      <mat-form-field appearance="legacy" style="width: 100%;">
        <mat-label>Ingrese su teléfono o celular</mat-label>
        <input matInput placeholder="2611223344" formControlName="telefono"
autocomplete="off">
        <mat-error *ngIf="form.controls['telefono'].errors">El
          teléfono ingresado no es válido</mat-error>
      </mat-form-field>
    </div>

    <!-- Input para obtener email del nuevo usuario -->
    <div class="my-3">
      <mat-form-field appearance="legacy" style="width: 100%;">
        <mat-label>Ingrese su email</mat-label>
        <input matInput placeholder="pat@example.com"
formControlName="correo" autocomplete="off">

```

```

        <mat-error *ngIf="form.controls['correo'].errors">El
            email ingresado no es válido</mat-error>
    </mat-form-field>
</div>

<!-- Input para obtener contraseña del nuevo usuario -->
<div class="my-3">
    <mat-form-field appearance="legacy" style="width: 100%;">
        <mat-label>Ingrese su contraseña</mat-label>
        <input matInput [type]="hide ? 'password' : 'text'"
placeholder="123456" formControlName="password"
            autocomplete="off">
        <button mat-icon-button matSuffix (click)="hide = !hide"
[attr.aria-label]='Hide password'"
            [attr.aria-pressed]="hide" type="button">
            <mat-icon>{{hide ? 'visibility_off' : 'visibility'}}</mat-icon>
        </button>
        <mat-error *ngIf="form.controls['password'].errors">La contraseña
debe tener al menos 6 caracteres</mat-error>
    </mat-form-field>
</div>

<!-- Input para obtener el rol que el nuevo usuario va a tener
en el sistema dependiendo de este se mostrara otro formulario
con datos específicos de cada rol -->
<div class="my-3">
    <mat-form-field appearance="legacy" style="width: 100%;">
        <mat-label>Tipo de usuario</mat-label>
        <mat-select formControlName="rol">
            <ng-container *ngFor="let rol of roles">
                <mat-option [value]=rol>
                    {{rol}}
                </mat-option>
            </ng-container>
        </mat-select>
    </mat-form-field>
</div>

<!-- Dependiendo del rol elegido se carga un formulario específico -->
<div class="my-3" *ngIf="form.controls['rol'].valid">
    <ng-container
*ngIf="form.controls['rol'].value.toUpperCase()=='DEPORTISTA'">
        <app-deportista [parentSubject]="parentSubject" [paises]=paises
[provincias]=provincias
            [departamentos]=departamentos
(cambioDatos)="cambioDatosDelRol($event)"></app-deportista>
    </ng-container>
    <ng-container
*ngIf="form.controls['rol'].value.toUpperCase()=='ORGANIZADOR'">
        <app-organizador [parentSubject]="parentSubject" [paises]=paises
[provincias]=provincias
            (cambioDatos)="cambioDatosDelRol($event)"></app-organizador>
    </ng-container>

```

```

        <ng-container
*ngIf="form.controls['rol'].value.toUpperCase()=='ARBITRO'">
            <app-arbitro [parentSubject]="parentSubject" [paises]=paises
[provincias]=provincias
                (cambioDatos)="cambioDatosDelRol($event)"></app-arbitro>
        </ng-container>
        <ng-container *ngIf="form.controls['rol'].value.toUpperCase()=='CLUB'">
            <app-club [parentSubject]="parentSubject" [paises]=paises
[provincias]=provincias
                [departamentos]=departamentos [localidades]=localidades
(cambioDatos)="cambioDatosDelRol($event)">
            </app-club>
        </ng-container>
        <ng-container
*ngIf="form.controls['rol'].value.toUpperCase()=='PATROCINADOR'">
            <app-patrocinador [parentSubject]="parentSubject"
(cambioDatos)="cambioDatosDelRol($event)">
            </app-patrocinador>
        </ng-container>
    </div>

    <!-- Botones para volver al login o para registrar el nuevo usuario -->
    <div class="row mt-3">
        <div class="col-12 text-center mb-2">
            <button type="submit" mat-raised-button color="primary" style="width:
100%;"
                (click)="enviarInformacion()">Registrarse</button>
        </div>
        <div class="col-12 text-center mb-2">
            <button type="button" mat-stroked-button color="primary"
style="width: 100%;"
                (click)="volver()">Volver</button>
        </div>
    </div>
</form>
</ng-template>

```

## Componente padre SignUp.component.ts

```

import { Component, OnInit } from '@angular/core';
import { FormControl, FormGroup, Validators } from '@angular/forms';
import { Router } from '@angular/router';
import { blankSpaceValidator } from '@helpers/blankspaceValidator';
import { DialogsService } from '@services/dialogs.service';
import { GlobalService } from '../services/global.service';
import { paisClass } from '../models/pais.model';
import { provinciaClass } from '../models/provincia.model';
import { departamentoClass } from '../models/departamento.model';
import { localidadClass } from '../models/localidad.model';
import { Subject } from 'rxjs';
import { AuthService } from '../services/auth.service';

```

```

@Component({
  selector: 'app-signup',
  templateUrl: './signup.component.html',
  styleUrls: ['./signup.component.scss'],
})
export class SignupComponent implements OnInit {
  hide: boolean = true;

  // Formulario con datos basicos de todos Los usuarios
  form: FormGroup = new FormGroup({
    nombre: new FormControl('', [Validators.required, blankSpaceValidator]),
    telefono: new FormControl('', [
      Validators.required,
      blankSpaceValidator,
      Validators.minLength(6),
      Validators.maxLength(11),
      Validators.pattern('^[0-9]*$'),
    ]),
    img: new FormControl('', []),
    rol: new FormControl('', [Validators.required, blankSpaceValidator]),
    correo: new FormControl('', [
      Validators.required,
      Validators.email,
      blankSpaceValidator,
    ]),
    password: new FormControl('', [Validators.required, blankSpaceValidator]),
  });

  // Datos necesarios para el signup, se usan en Los componenetes hijos
  roles: Array<any> = [];
  paises: Array<paisClass> = [];
  provincias: Array<provinciaClass> = [];
  departamentos: Array<departamentoClass> = [];
  localidades: Array<localidadClass> = [];

  datosRol: any = null; //Datos que vienen de componente hijo
  estadoDatosRol: boolean = false; //Estado del formulario del componente hijo

  parentSubject: Subject<any> = new Subject(); //Para informarLe a Los child components
  que se submiteo

  constructor(
    private router: Router,
    private globalService: GlobalService,
    private dialogsService: DialogsService,
    private authService: AuthService
  ) {}

  // Cuando inicia se busca La data en La bd
  ngOnInit(): void {
    this.getData();
  }
}

```

```
// Se busca toda la data necesaria para poder realizar el sigup correctamente
getData() {
  this.dialogsService.cargando('Buscando datos necesarios para registro');
  this.globalService.getDatosSignUp().subscribe(
    (respuesta) => {
      this.dialogsService.cerrar();
      this.roles = respuesta.roles;
      this.países = respuesta.países;
      this.provincias = respuesta.provincias;
      this.departamentos = respuesta.departamentos;
      this.localidades = respuesta.localidades;
    },
    (error) => {
      this.dialogsService.cerrar();
      this.dialogsService.toastError(
        'Se produjo un error, vuelva a intentarlo más tarde'
      );
      this.router.navigateByUrl('inicio');
    }
  );
}

// Cuando se produce un cambio en el componente hijo nos llega la data y el estado del formulario
cambioDatosDelRol(respuesta: any) {
  this.datosRol = respuesta.datos;
  this.estadoDatosRol = respuesta.estado;
}

// Metodo para registrar usuario
enviarInformacion() {
  this.parentSubject.next(true); //Se le informa al componente hijo que se intento registrar el usuario

  // Se comprueba que no haya errores en el formulario del componente padre ni del componente hijo
  if (this.form.invalid || !this.estadoDatosRol) {
    return;
  }

  // Se crea un carte informandole al usuario que espere
  this.dialogsService.cargando('Creando cuenta por favor espere');

  // Se agrupa toda la data que se le debe enviar al Backend
  let usuarioData = this.form.getRawValue();
  let data: any = {
    nombre: usuarioData.nombre,
    telefono: usuarioData.telefono,
    fechaInicioVig: new Date(),
    correo: usuarioData.correo,
    password: usuarioData.password,
    img: usuarioData.img,
  }
}
```

```
deportista: null,  
organizador: null,  
club: null,  
arbitro: null,  
patrocinador: null,  
rol: usuarioData.rol,  
};  
  
// Dependiendo del rol se debe agregar diferentes datos  
switch (usuarioData.rol) {  
  case 'DEPORTISTA':  
    data.apellido = this.datosRol.apellido;  
    data.departamento = this.datosRol.idDepartamento;  
    data.DNI = this.datosRol.DNI;  
    data.fechaInicioVig = new Date();  
    data.fechaNacimiento = this.datosRol.fechaNacimiento;  
    data.usuario = null;  
    data.deportes = [];  
    break;  
  
  case 'ORGANIZADOR':  
    data.provincia = this.datosRol.idProvincia;  
    data.CUIT = this.datosRol.CUIT;  
    data.tipoOrganizador = 'GRATUITO';  
    break;  
  
  case 'CLUB':  
    data.domicilio = this.datosRol.domicilio;  
    data.latitud = this.datosRol.latitud;  
    data.longitud = this.datosRol.longitud;  
    data.localidad = this.datosRol.idLocalidad;  
    data.CUIT = this.datosRol.CUIT;  
    break;  
  
  case 'ARBITRO':  
    data.provincia = this.datosRol.idProvincia;  
    data.DNI = this.datosRol.DNI;  
    data.apellido = this.datosRol.apellido;  
    data.deporte = this.datosRol.idDeporte;  
    break;  
  
  case 'PATROCINADOR':  
    data.apellido = this.datosRol.apellido;  
    data.CUIT = this.datosRol.CUIT;  
    break;  
  
  default:  
    break;  
}  
  
// Se envia La información al BackEnd  
this.authService.signup(data).subscribe(  
  (respuesta) => {
```

```
// Si se pudo crear correctamente, se cierra el cargando,  
//se informa al usuario que vaya a su mail y se lo redirige al login  
this.dialogsService.cerrar();  
this.dialogsService.toastSuccess(  
  'Se creo usuario correctamente, por favor verifique la cuenta en su mail'  
);  
this.router.navigateByUrl('login');  
},  
(error) => {  
  // Si se produce un error se le informa el mismo al usuario  
  let message = 'Se produjo un error,vuelva a intentarlo más tarde';  
  if (  
    error.error.message == 'Ya existe un usuario con el email ingresado'  
  ) {  
    message = 'Ya existe un usuario con el email ingresado';  
  }  
  this.dialogsService.cerrar();  
  this.dialogsService.showError(message);  
}  
);  
}
```

```
// Funcion para volver al Login, usada por el botón volver  
volver() {  
  this.router.navigate(['login']);  
}
```

### Componente hijo deportista.component.html

```
<!-- Formulario para obtener datos especificos del deportista -->  
<form [formGroup]="form">  
  
  <!-- Input para obtener apellido del nuevo usuario -->  
  <div class="my-3">  
    <mat-form-field appearance="legacy" style="width: 100%;">  
      <mat-label>Ingrese su apellido</mat-label>  
      <input matInput placeholder="Perez" formControlName="apellido"  
autocomplete="off">  
      <mat-error *ngIf="form.controls['apellido'].errors">El  
        apellido es obligatorio</mat-error>  
    </mat-form-field>  
  </div>  
  
  <!-- Input para obtener fecha de nacimiento del nuevo usuario -->  
  <div class="my-3">  
    <mat-form-field appearance="legacy" style="width: 100%;">  
      <mat-label>Fecha de nacimiento</mat-label>  
      <input matInput [matDatepicker]="picker" formControlName="fechaNacimiento"  
autocomplete="off">  
      <mat-datepicker-toggle matSuffix [for]="picker"></mat-datepicker-toggle>  
      <mat-datepicker #picker></mat-datepicker>  
    </mat-form-field>  
  </div>  
</form>
```

```

    </mat-form-field>
</div>

<!-- Input para obtener DNI del nuevo usuario -->
<div class="my-3">
  <mat-form-field appearance="legacy" style="width: 100%;">
    <mat-label>Ingrese su DNI</mat-label>
    <input matInput placeholder="11222333" formControlName="DNI"
autocomplete="off">
    <mat-error *ngIf="form.controls['DNI'].errors">El
      DNI es obligatorio</mat-error>
    </mat-form-field>
  </div>

<!-- Input para obtener pais del nuevo usuario -->
<div class="my-3">
  <mat-form-field appearance="legacy" style="width: 100%;">
    <mat-label>País</mat-label>
    <mat-select formControlName="pais">
      <ng-container *ngFor="let pais of paises">
        <mat-option [value]=pais>
          {{pais.nombre}}
        </mat-option>
      </ng-container>
    </mat-select>
    <mat-error *ngIf="form.controls['pais'].errors">El
      país es obligatorio</mat-error>
    </mat-form-field>
  </div>

<!-- Input para obtener provincia del nuevo usuario se usa un filtro
para mostrar solo las provincias del pais seleccionado -->
<div class="my-3">
  <mat-form-field appearance="legacy" style="width: 100%;">
    <mat-label>Provincia</mat-label>
    <mat-select formControlName="provincia">
      <ng-container *ngFor="let provincia of provincias |
filterProvinciaByPais:paisSeleccionado">
        <mat-option [value]=provincia>
          {{provincia.nombre}}
        </mat-option>
      </ng-container>
    </mat-select>
    <mat-error *ngIf="form.controls['provincia'].errors">La provincia es
obligatoria</mat-error>
    </mat-form-field>
  </div>

<!-- Input para obtener departamento del nuevo usuario se usa un filtro
para mostrar solo los departamentos de la provincia seleccionada -->
<div class="my-3">
  <mat-form-field appearance="legacy" style="width: 100%;">
    <mat-label>Departamento</mat-label>

```



```

    <mat-select formControlName="departamento">
      <ng-container
        *ngFor="let departamento of departamentos |
filterDepartamentoByProvincia:provinciaSeleccionada">
        <mat-option [value]=departamento>
          {{departamento.nombre}}
        </mat-option>
      </ng-container>
    </mat-select>
    <mat-error *ngIf="form.controls['departamento'].errors">El
      departamento es obligatorio</mat-error>
  </mat-form-field>
</div>
</form>

```

### Componente hijo deportista.component.ts

```

import {
  Component,
  EventEmitter,
  Input,
  OnChanges,
  OnDestroy,
  OnInit,
  Output,
} from '@angular/core';
import { FormControl, FormGroup, Validators } from '@angular/forms';
import { blankSpaceValidator } from '@helpers/blankSpaceValidator';
import { Subject, Subscription } from 'rxjs';

@Component({
  selector: 'app-deportista',
  templateUrl: './deportista.component.html',
  styleUrls: ['./deportista.component.scss'],
})
export class DeportistaComponent implements OnInit, OnDestroy, OnChanges {
  private subscriptionForm: Subscription = new Subscription();
  private subscriptionPais: Subscription = new Subscription();
  private subscriptionProvincia: Subscription = new Subscription();
  private subscriptionParentSubject: Subscription = new Subscription();
  hide: boolean = true;

  // Se crea formulario específico para deportista
  form: FormGroup = new FormGroup({
    apellido: new FormControl('', [Validators.required, blankSpaceValidator]),
    fechaNacimiento: new FormControl('', [
      Validators.required,
      blankSpaceValidator,
    ]),
    DNI: new FormControl('', [Validators.required, blankSpaceValidator]),
    pais: new FormControl('', [Validators.required, blankSpaceValidator]),
    provincia: new FormControl('', [Validators.required, blankSpaceValidator]),
    departamento: new FormControl('', [

```

```
Validators.required,  
    blankSpaceValidator,  
  ]),  
});  
  
// Se reciben datos necesarios desde el componente padre  
@Input() paises: Array<any> = [];  
@Input() provincias: Array<any> = [];  
@Input() departamentos: Array<any> = [];  
@Input() parentSubject!: Subject<any>;  
@Input() usuarioViejo!: any;  
@Output() cambioDatos = new EventEmitter<any>();  
  
paisSeleccionado: string = '';  
provinciaSeleccionada: string = '';  
  
constructor() {}  
  
ngOnInit(): void {  
  // Me suscribo a este evento para saber si el usuario toco registrarse desde el  
componente padre  
  // y revisar que no haya errores en este componente hijo  
  this.subscriptionParentSubject = this.parentSubject.subscribe((resp) => {  
    if (resp) {  
      this.form.markAllAsTouched();  
    }  
  });  
  // LLamo a funciones para escuchar cambios en el formulario  
  this.escucharCambiosEnForm();  
  this.cambiosEnPais();  
  this.cambiosEnProvincia();  
}  
  
// Este componente se reutiliza en módulo perfil, esto es para cargar datos anteriores  
ngOnChanges() {  
  if (  
    this.paises.length > 0 &&  
    this.provincias.length > 0 &&  
    this.departamentos.length > 0 &&  
    this.usuarioViejo != null  
  ) {  
    this.cargarDatosAModificar();  
  }  
}  
  
// Se cargan datos del usuario anteriores  
cargarDatosAModificar() {  
  this.form.controls['apellido'].setValue(  
    this.usuarioViejo.deportista.apellido  
  );  
  this.form.controls['DNI'].setValue(this.usuarioViejo.deportista.DNI);  
  this.form.controls['fechaNacimiento'].setValue(  
    this.usuarioViejo.deportista.fechaNacimiento
```

```
);

let departamento = this.departamentos.find(
  (x) => x._id == this.usuarioViejo.deportista.departamento
);
let provincia = this.provincias.find(
  (x) => x._id == departamento?.provincia
);
let pais = this.países.find((x) => x._id == provincia?.pais);

if (pais && provincia && departamento) {
  this.form.controls['pais'].setValue(pais);
  this.form.controls['provincia'].setValue(provincia);
  this.form.controls['departamento'].setValue(departamento);
}
}

// Cada vez que se produce un cambio en el formulario se envía la información al
componente padre
escucharCambiosEnForm() {
  this.subscriptionForm = this.form.valueChanges.subscribe(() => {
    let data = this.form.getRawValue();
    let datos = {
      datos: {
        apellido: data.apellido,
        idDepartamento: data.departamento._id,
        DNI: data.DNI,
        fechaNacimiento: data.fechaNacimiento?._d || new Date(),
      },
      estado: this.form.valid,
    };
    this.cambioDatos.emit(datos);
  });
}

// Cada vez que se cambia el país, elimina la provincia y departamento seteados y se
actualiza el filtro
cambiosEnPais() {
  this.subscriptionPais = this.form.controls['pais'].valueChanges.subscribe(
    () => {
      this.paisSeleccionado = this.form.getRawValue().pais._id;
      this.provinciaSeleccionada = '';
      this.form.controls['provincia'].setValue('');
      this.form.controls['departamento'].setValue('');
    }
  );
}

// Cada vez que se cambia el país, elimina el departamento seteado y se actualiza el
filtro
cambiosEnProvincia() {
  this.subscriptionProvincia = this.form.controls[
    'provincia'
```

```

    ].valueChanges.subscribe(() => {
      this.provinciaSeleccionada = this.form.getRawValue().provincia._id;
      this.form.controls['departamento'].setValue('');
    });
  }

  // Me desuscribo de las subscripciones para evitar pérdidas de memoria
  ngOnDestroy(): void {
    this.subscriptionForm.unsubscribe();
    this.subscriptionPais.unsubscribe();
    this.subscriptionProvincia.unsubscribe();
    this.subscriptionParentSubject.unsubscribe();
  }
}

```

### Service que se llama desde componente padre auth.service.ts

```

// Se llama al back y se devuelve la respuesta como observable
signup(datos: any) {
  return this.http
    .post<any>(`${environment.url}/auth/signup`, datos)
    .pipe(take(1));
}

```

### HU N°2 SignUp Club

#### Planilla de documentación

<b>Nombre de la HU</b>	Sign up Club
<b>Módulo seleccionado</b>	Seguridad
<b>Descripción breve de la HU</b>	Como potencial usuario con rol club quiero crear una cuenta para poder identificarme en el sistema.
<b>Actor</b>	Usuario
<b>Prioridad</b>	A
<b>Programador BackEnd</b>	Ignacio Starkloff
<b>Programador FrontEnd</b>	Juan Riera
<b>Tester</b>	Jose Aranciba
<b>Aprobado por</b>	Miguel Del Corso

<b>Denominación del programa</b>	<b>backend</b> “signup.controller.js” “signup.service.js” “signup.helper.js” <b>frontend</b> “SignUp.component.ts” “SignUp.component.html”
<b>Fecha inicio Programación</b>	09/08/2021
<b>Fecha finalización de programación</b>	13/08/2021
<b>Base de datos</b>	MongoDB Redis
<b>Colecciones involucradas</b>	Usuario Estado Usuario Estado Club Rol UsuarioRol Localidad
<b>Historias de usuarios relacionadas</b>	HU 6 - Login HU 2 - SignUp deportista
<b>Casos de prueba relacionados con resultados</b>	
<p> Crear usuario club con el campo cuit con el campo vacío=&gt; PASS  Crear usuario club con el campo cuit con Un cuit correcto=&gt; PASS  Crear usuario club con el campo cuit con un cuit con ceros=&gt; PASS  Crear usuario club con el campo cuit con un cuit con coma=&gt; PASS  Crear usuario club con el campo cuit con un cuit con punto=&gt; PASS  Crear usuario club con el campo cuit con un cuit negativo=&gt; PASS  Crear usuario club con el campo cuit con una expresión matemática=&gt; PASS  Crear usuario club con el campo cuit con una cadena de caracteres alfabéticos=&gt; PASS  Crear usuario club con el campo cuit con un carácter alfabético=&gt; PASS  Crear usuario club con el campo cuit con un carácter alfabético entre enteros=&gt; PASS  Crear usuario club con el campo cuit con un carácter especial=&gt; PASS  Crear usuario club con el campo cuit con null=&gt; PASS  Crear usuario club con el campo cuit con Nan=&gt; PASS  Crear usuario club con el campo cuit con NaN=&gt; PASS  Crear usuario club con el campo cuit con Na=&gt; PASS  Crear usuario club con el campo cuit con false=&gt; PASS  Crear usuario club con el campo cuit con true=&gt; PASS  Crear usuario club con el campo domicilio con un array vacío=&gt; PASS  Crear usuario club con el campo domicilio con caracteres especiales=&gt; PASS  Crear usuario club con el campo domicilio con Un entero =&gt; PASS  Crear usuario club con el campo domicilio con Un número con coma =&gt; PASS  Crear usuario club con el campo domicilio con Un número con punto =&gt; PASS  Crear usuario club con el campo domicilio con Un espacio en blanco=&gt; PASS </p>	

Crear usuario club con el campo domicilio con un json mal formulado => PASS  
Crear usuario club con el campo domicilio con los caracteres: \" => PASS  
Crear usuario club con el campo domicilio con Dos caracteres: \ => PASS  
Crear usuario club con el campo latitud con el campo vacío=> PASS  
Crear usuario club con el campo latitud con Un número correcto=> PASS  
Crear usuario club con el campo latitud con el número cero=> PASS  
Crear usuario club con el campo latitud con un número con coma=> PASS  
Crear usuario club con el campo latitud con un número con punto=> PASS  
Crear usuario club con el campo latitud con un número negativo=> PASS  
Crear usuario club con el campo latitud con una expresión matemática=> PASS  
Crear usuario club con el campo latitud con una cadena de caracteres alfabeticos=> PASS  
Crear usuario club con el campo latitud con un carácter alfabético=> PASS  
Crear usuario club con el campo latitud con un carácter alfabético entre enteros=> PASS  
Crear usuario club con el campo latitud con un carácter especial=> PASS  
Crear usuario club con el campo latitud con más de 100 cifras=> PASS  
Crear usuario club con el campo latitud con null=> PASS  
Crear usuario club con el campo latitud con Nan=> PASS  
Crear usuario club con el campo latitud con NaN=> PASS  
Crear usuario club con el campo latitud con Na=> PASS  
Crear usuario club con el campo latitud con false=> PASS  
Crear usuario club con el campo latitud con true=> PASS  
Crear usuario club con el campo longitud con el campo vacío=> PASS  
Crear usuario club con el campo longitud con Un número correcto=> PASS  
Crear usuario club con el campo longitud con el número cero=> PASS  
Crear usuario club con el campo longitud con un número con coma=> PASS  
Crear usuario club con el campo longitud con un número con punto=> PASS  
Crear usuario club con el campo longitud con un número negativo=> PASS  
Crear usuario club con el campo longitud con una expresión matemática=> PASS  
Crear usuario club con el campo longitud con una cadena de caracteres alfabeticos=> PASS  
Crear usuario club con el campo longitud con un carácter alfabético=> PASS  
Crear usuario club con el campo longitud con un carácter alfabético entre enteros=> PASS  
Crear usuario club con el campo longitud con un carácter especial=> PASS  
Crear usuario club con el campo longitud con más de 100 cifras=> PASS  
Crear usuario club con el campo longitud con null=> PASS  
Crear usuario club con el campo longitud con Nan=> PASS  
Crear usuario club con el campo longitud con NaN=> PASS  
Crear usuario club con el campo longitud con Na=> PASS  
Crear usuario club con el campo longitud con false=> PASS  
Crear usuario club con el campo longitud con true=> PASS  
Crear usuario club con el campo usuario con el campo vacío=> PASS  
Crear usuario club con el campo usuario con un usuario inexistente=> PASS  
Crear usuario club con el campo usuario con un usuario no habilitado=> PASS  
Crear usuario club con el campo usuario con un json mal formulado => PASS

## Documentación del BackEnd

```
const crearModeloClub = async (usuario) => {
  // Datos que son necesarios para crear el modelo del club
  const { domicilio, latitud, longitud, localidad, CUIT, _id } = usuario;
  if (localidad && domicilio && latitud && longitud && _id) {
    // Buscamos la localidad en la base de datos
    const modeloLocalidad = await findLocalidadByIdService(localidad);
    // Creamos modelo del club
    const modeloClub = new Club({
      CUIT: CUIT,
      domicilio: domicilio,
      latitud: latitud,
      longitud: longitud,
      localidad: modeloLocalidad._id,
      usuario: _id,
    });
    // Guardamos al club en la base de datos
    await saveClubService(modeloClub);
    return modeloClub;
  } else {
    throw new BadRequest(
      "El formato enviado no es el adecuado para crear un club"
    );
  }
};
```

## Pantallas

**SPORTINEV**

### Registrarse

Ingrese su nombre

Ingrese su teléfono o celular

Ingrese su email

Ingrese su contraseña

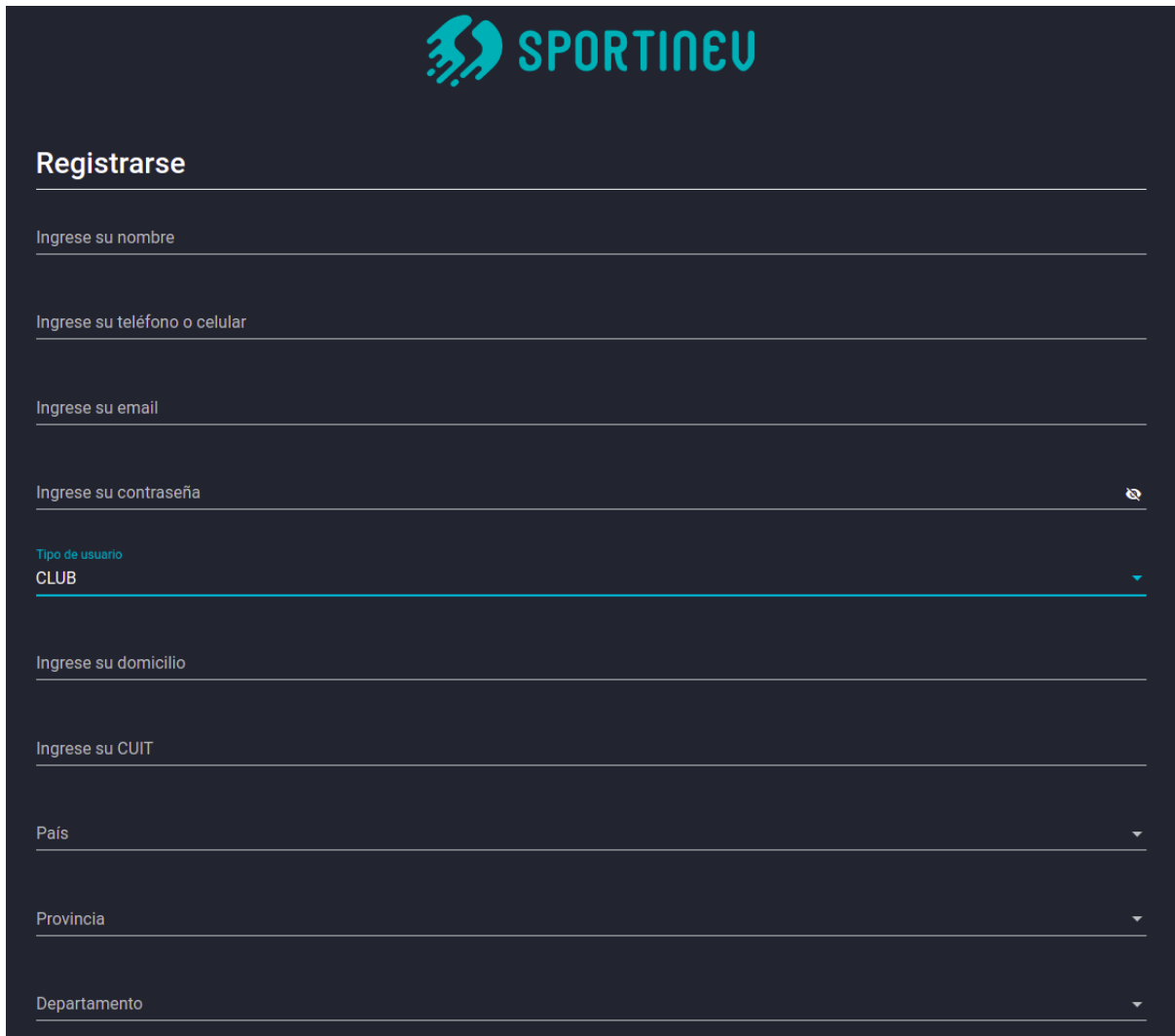
Tipo de usuario

**Registrarse**

Volver

Figura 254: Pantalla de registrarse





The image shows a registration form for 'SPORTINEV' on a dark background. The form is titled 'Registrarse' and contains several input fields and dropdown menus. The fields are: 'Ingrese su nombre', 'Ingrese su teléfono o celular', 'Ingrese su email', 'Ingrese su contraseña' (with a toggle icon), 'Tipo de usuario' (a dropdown menu currently showing 'CLUB'), 'Ingrese su domicilio', 'Ingrese su CUIT', 'País', 'Provincia', and 'Departamento' (all dropdown menus).

Figura 255: Pantalla de registrarse como club

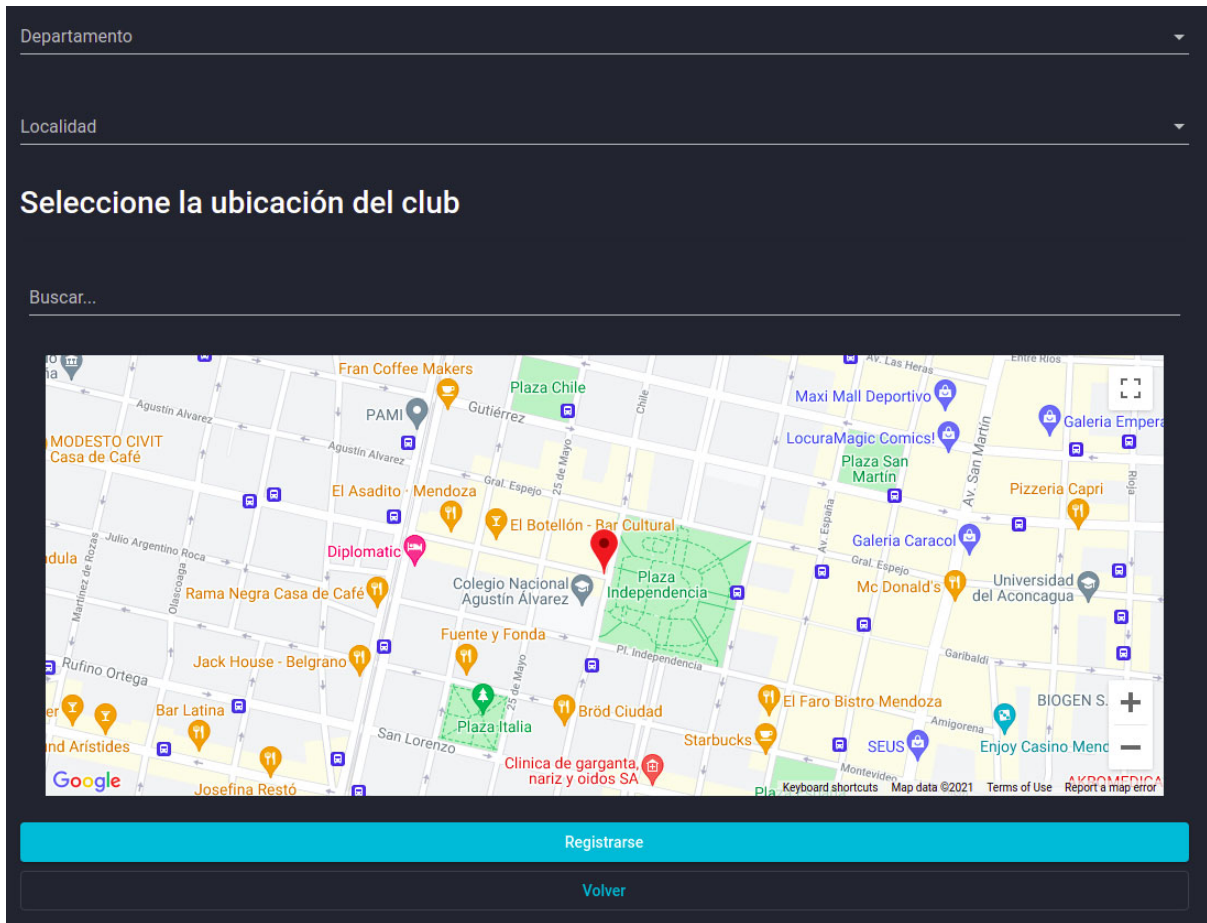


Figura 256: Pantalla de selección de ubicación del club

## Documentación del FrontEnd

Debido a que para la siguiente historia de usuario se utilizan los mismo componentes padres y services que fueron detallados anteriormente en SignUp Deportista, solo se detalla el componente hijo que es llamado una vez que se selecciona el tipo de rol Club.

### Componente hijo club.component.html

```
<!-- Formulario especifico para usuario tipo club -->
<form [formGroup]="form">

  <!-- Input para obtener domicilio del nuevo usuario -->
  <div class="my-3">
    <mat-form-field appearance="legacy" style="width: 100%;">
      <mat-label>Ingrese su domicilio</mat-label>
      <input matInput placeholder="Godoy Cruz" formControlName="domicilio"
autocomplete="off">
      <mat-error *ngIf="form.controls['domicilio'].errors">El
        domicilio es obligatorio</mat-error>
    </mat-form-field>
  </div>

  <!-- Input para obtener CUIT del nuevo usuario -->
  <div class="my-3">
    <mat-form-field appearance="legacy" style="width: 100%;">
      <mat-label>Ingrese su CUIT</mat-label>
      <input matInput placeholder="20112223335" formControlName="CUIT"
autocomplete="off">
      <mat-error *ngIf="form.controls['CUIT'].errors">El
        CUIT es obligatorio</mat-error>
    </mat-form-field>
  </div>

  <!-- Input para obtener oais del nuevo usuario -->
  <div class="my-3">
    <mat-form-field appearance="legacy" style="width: 100%;">
      <mat-label>País</mat-label>
      <mat-select formControlName="pais">
        <ng-container *ngFor="let pais of paises">
          <mat-option [value]=pais>
            {{pais.nombre}}
          </mat-option>
        </ng-container>
      </mat-select>
    </mat-form-field>
  </div>

  <!-- Input para obtener provincia del nuevo usuario se usa un filtro
  para mostrar solo las provincias del pais seleccionado -->
  <div class="my-3">
    <mat-form-field appearance="legacy" style="width: 100%;">
      <mat-label>Provincia</mat-label>
      <mat-select formControlName="provincia">
        <ng-container *ngFor="let provincia of provincias |
filterProvinciaByPais:paisSeleccionado">
          <mat-option [value]=provincia>
```

```

                {{provincia.nombre}}
            </mat-option>
        </ng-container>

    </mat-select>
</mat-form-field>
</div>

<!-- Input para obtener departamento del nuevo usuario se usa un filtro
para mostrar solo los departamentos de la provincia seleccionada -->
<div class="my-3">
    <mat-form-field appearance="legacy" style="width: 100%;">
        <mat-label>Departamento</mat-label>
        <mat-select formControlName="departamento">
            <ng-container
                *ngFor="let departamento of departamentos |
filterDepartamentoByProvincia:provinciaSeleccionada">
                <mat-option [value]=departamento>
                    {{departamento.nombre}}
                </mat-option>
            </ng-container>
        </mat-select>
    </mat-form-field>
</div>

<!-- Input para obtener localidad del nuevo usuario se usa un filtro
para mostrar solo las localidades del departamento seleccionado -->
<div class="my-3">
    <mat-form-field appearance="legacy" style="width: 100%;">
        <mat-label>Localidad</mat-label>
        <mat-select formControlName="localidad">
            <ng-container
                *ngFor="let localidad of localidades |
filterLocalidadByDepartamento:departamentoSeleccionado">
                <mat-option [value]=localidad>
                    {{localidad.nombre}}
                </mat-option>
            </ng-container>
        </mat-select>
    </mat-form-field>
</div>

<!-- Mapa para que el usuario seleccione donde se encuentra el club
Se usan dos componentes hijos para esto app-search-autocomplete para predecir la
ubicacion con google
y app-map donde se muestra lo que va seleccionando el usuario en un mapa
-->
<h3>Seleccione la ubicación del club</h3>
<hr>
<div class="p-2">
    <app-search-autocomplete
(setCoordenadasBusqueda)="setCoordenadasBusqueda(($event))"></app-search-autocomplete>
    <div class="col-12">

```

```
        <app-map [parentSubject]=parentSubjectAEnviar
(setCoordenadas)="setCoordenadas(($event))"></app-map>
    </div>
</div>

</form>
```

### Componente hijo club.component.ts

```
import {
  Component,
  EventEmitter,
  Input,
  OnChanges,
  OnDestroy,
  OnInit,
  Output,
} from '@angular/core';
import { FormControl, FormGroup, Validators } from '@angular/forms';
import { blankSpaceValidator } from '@helpers/blankSpaceValidator';
import { Subject, Subscription } from 'rxjs';

@Component({
  selector: 'app-club',
  templateUrl: './club.component.html',
  styleUrls: ['./club.component.scss'],
})
export class ClubComponent implements OnInit, OnDestroy, OnChanges {
  private subscriptionForm: Subscription = new Subscription();
  private subscriptionPais: Subscription = new Subscription();
  private subscriptionProvincia: Subscription = new Subscription();
  private subscriptionDepartamento: Subscription = new Subscription();
  private subscriptionParentSubject: Subscription = new Subscription();
  hide: boolean = true;

  // Formulario especifico para usuarios de tipo club
  form: FormGroup = new FormGroup({
    CUIT: new FormControl('', [Validators.required, blankSpaceValidator]),
    domicilio: new FormControl('', [Validators.required, blankSpaceValidator]),
    pais: new FormControl('', [Validators.required, blankSpaceValidator]),
    provincia: new FormControl('', [Validators.required, blankSpaceValidator]),
    departamento: new FormControl('', [
      Validators.required,
      blankSpaceValidator,
    ]),
    localidad: new FormControl('', [Validators.required, blankSpaceValidator]),
    latitud: new FormControl('', [Validators.required, blankSpaceValidator]),
    longitud: new FormControl('', [Validators.required, blankSpaceValidator]),
  });

  // Datos obtenidos del componente padre
  @Input() paises: Array<any> = [];
  @Input() provincias: Array<any> = [];
```

```
@Input() departamentos: Array<any> = [];  
@Input() localidades: Array<any> = [];  
@Input() parentSubject!: Subject<any>;  
@Input() usuarioViejo: any = null;  
@Output() cambioDatos = new EventEmitter<any>();  
  
paisSeleccionado: string = '';  
provinciaSeleccionada: string = '';  
departamentoSeleccionado: string = '';  
  
parentSubjectAEnviar: Subject<any> = new Subject();  
constructor() {}  
  
ngOnInit(): void {  
    // Me suscribo a este evento para saber si el usuario toco registrarse desde el  
    // componente padre  
    // y revisar que no haya errores en este componente hijo  
    this.subscriptionParentSubject = this.parentSubject.subscribe((resp) => {  
        if (resp) {  
            this.form.markAllAsTouched();  
        }  
    });  
    // LLamo a funciones para escuchar cambios en el formulario  
    this.escucharCambiosEnForm();  
    this.cambiosEnPais();  
    this.cambiosEnProvincia();  
    this.cambiosEnDepartamento();  
}  
  
// Este componente se reutiliza en modulo perfil, esto es para cargar datos anteriores  
ngOnChanges() {  
    if (  
        this.paises.length > 0 &&  
        this.provincias.length > 0 &&  
        this.departamentos.length > 0 &&  
        this.localidades.length > 0 &&  
        this.usuarioViejo != null  
    ) {  
        this.cargarDatosAModificar();  
    }  
}  
  
// Se cargan datos del usuario anteriores  
cargarDatosAModificar() {  
    this.form.controls['domicilio'].setValue(this.usuarioViejo.club.domicilio);  
    let localidad = this.localidades.find(  
        (x) => x._id == this.usuarioViejo.club.localidad  
    );  
    console.log(localidad);  
  
    let departamento = this.departamentos.find(  
        (x) => x._id == localidad?.departamento  
    );  
};
```

```
let provincia = this.provincias.find(
  (x) => x._id == departamento?.provincia
);
let pais = this.países.find((x) => x._id == provincia?.pais);

if (pais && provincia && departamento && localidad) {
  this.form.controls['pais'].setValue(pais);
  this.form.controls['provincia'].setValue(provincia);
  this.form.controls['departamento'].setValue(departamento);
  this.form.controls['localidad'].setValue(localidad);
}

this.form.controls['CUIT'].setValue(this.usuarioViejo.club.CUIT);
this.setCoordenadasBusqueda(
  new google.maps.LatLng(
    this.usuarioViejo.club.latitud,
    this.usuarioViejo.club.longitud
  )
);
}

// Cada vez que se produce un cambio en el formulario se envía la información al
componente padre
escucharCambiosEnForm() {
  this.subscriptionForm = this.form.valueChanges.subscribe(() => {
    let data = this.form.getRawValue();
    console.log(data);

    let datos = {
      datos: {
        domicilio: data.domicilio,
        latitud: data.latitud,
        longitud: data.longitud,
        idLocalidad: data.localidad._id,
        CUIT: data.CUIT,
      },
      estado: this.form.valid,
    };
    this.cambioDatos.emit(datos);
  });
}

// Cada vez que se cambia el país, elimina la provincia y departamento seteados y se
actualiza el filtro
cambiosEnPais() {
  this.subscriptionPais = this.form.controls['pais'].valueChanges.subscribe(
    () => {
      this.paisSeleccionado = this.form.getRawValue().pais._id;
      this.provinciaSeleccionada = '';
      this.departamentoSeleccionado = '';
      this.form.controls['provincia'].setValue('');
      this.form.controls['departamento'].setValue('');
      this.form.controls['localidad'].setValue('');
    }
  );
}
```

```
    }  
  );  
}  
  
// Cada vez que se cambia el provincia, elimina el departamento seteado y se actualiza  
el filtro  
cambiosEnProvincia() {  
  this.subscriptionProvincia = this.form.controls[  
    'provincia'  
  ].valueChanges.subscribe(() => {  
    this.provinciaSeleccionada = this.form.getRawValue().provincia._id;  
    this.departamentoSeleccionado = '';  
    this.form.controls['departamento'].setValue('');  
    this.form.controls['localidad'].setValue('');  
  });  
}  
  
// Cada vez que se cambia el departamento, elimina la localidad seteada y se actualiza  
el filtro  
cambiosEnDepartamento() {  
  this.subscriptionProvincia = this.form.controls[  
    'departamento'  
  ].valueChanges.subscribe(() => {  
    this.departamentoSeleccionado = this.form.getRawValue().departamento._id;  
    this.form.controls['localidad'].setValue('');  
  });  
}  
  
// Me desuscribo de las subscripciones para evitar perdidas de memoria  
ngOnDestroy(): void {  
  this.subscriptionForm.unsubscribe();  
  this.subscriptionPais.unsubscribe();  
  this.subscriptionProvincia.unsubscribe();  
  this.subscriptionDepartamento.unsubscribe();  
  this.subscriptionParentSubject.unsubscribe();  
}  
  
// Cuando se producen cambios en el mapa se llama a este método para que las  
coordenadas se seteen en el formulario  
setCoordenadas(coordenadas: any) {  
  this.form.controls['latitud'].setValue(coordenadas.lat());  
  this.form.controls['longitud'].setValue(coordenadas.lng());  
}  
  
// Esta funcion se llama cuando se produce un cambio en el google search autocomplete  
// Sirve para informarle al componente hijo map que tiene que cambiar su ubicación  
setCoordenadasBusqueda(coordenadas: any) {  
  this.parentSubjectAEnviar.next(coordenadas);  
}  
}
```

## Componente hijo app-search-autocomplete.component.html



```
<!-- Input con autocompletado de direcciones de google -->
<mat-form-field appearance="legacy" style="width: 100%;">
  <mat-label>Buscar...</mat-label>
  <input matInput #search placeholder="Mendoza"
(keydown.enter)="$event.preventDefault()" autocomplete="off">
</mat-form-field>
```

### Componente hijo app-search-autocomplete.component.ts

```
import {
  Component,
  ElementRef,
  NgZone,
  OnInit,
  Output,
  ViewChild,
} from '@angular/core';
import { Store } from '@ngrx/store';
import { AppState } from '@store/app.reducers';
import { EventEmitter } from '@angular/core';

@Component({
  selector: 'app-search-autocomplete',
  templateUrl: './search-autocomplete.component.html',
  styleUrls: ['./search-autocomplete.component.scss'],
})
export class SearchAutocompleteComponent implements OnInit {
  @ViewChild('search') public searchElementRef!: ElementRef; //Se crea un referencia al
  input
  @Output() setCoordenadasBusqueda = new EventEmitter<any>();

  constructor(private ngZone: NgZone, private store: Store<AppState>) {}

  ngOnInit(): void {}

  ngAfterViewInit() {
    let autocomplete = new google.maps.places.Autocomplete(
      this.searchElementRef.nativeElement
    );
    autocomplete.setComponentRestrictions({ country: 'AR' });
    // Se subscribe a cambio en el input
    autocomplete.addListener('place_changed', () => {
      this.ngZone.run(() => {
        //Obtener el resultado de place
        let place: google.maps.places.PlaceResult = autocomplete.getPlace();

        // Verificar resultado
        if (place.geometry === undefined || place.geometry === null) {
          return;
        }
      })
    });
  }
}
```

```
//setear Latitud y Longitud
let coordenates = new google.maps.LatLng(
  place.geometry.location.lat(),
  place.geometry.location.lng()
);

//Se envia Logintud y Latitud al componente padre
this.setCoordenadasBusqueda.emit(coordenates);
});
});
}
}
```

### Componente hijo app-map.component.html

```
<!-- Mapa de google -->
<div class="mt-3">
  <div #mapContainer id="map"></div>
</div>
```

### Componente hijo app-map.component.ts

```
/// <reference types="@types/googLemaps" />
// La linea de arriba es para importar Los type de google
import {
  Component,
  ElementRef,
  EventEmitter,
  Input,
  OnInit,
  Output,
  ViewChild,
} from '@angular/core';
import { Subject, Subscription } from 'rxjs';
import { OnDestroy } from '@angular/core';

@Component({
  selector: 'app-map',
  templateUrl: './map.component.html',
  styleUrls: ['./map.component.scss'],
})
export class MapComponent implements OnInit, OnDestroy {
  private subscriptionGoogle: Subscription = new Subscription();
  @Input() parentSubject!: Subject<any>; //Subject que nos permite escuchar cambios que nos pide el componente padre
  @Output() setCoordenadas = new EventEmitter(); //EventEmmitter para informar a componente padre de cambios
```

```
coordenates: google.maps.LatLng = new google.maps.LatLng(  
  -32.8894587,  
  -68.8458386  
); //Coordenadas iniciales  
map: google.maps.Map | null = null;  
marker: google.maps.Marker | null = null;  
  
mapOptions: google.maps.MapOptions = {  
  zoom: 16,  
  mapTypeId: 'roadmap',  
  streetViewControl: false,  
  clickableIcons: false,  
  mapTypeControl: false,  
}; //Configuración de mapa inicial  
  
@ViewChild('mapContainer', { static: false })  
gmap: ElementRef | null = null;  
constructor() {}  
  
ngOnInit(): void {  
  // No suscribimos a las peticiones que nos hace componente padre  
  this.subscriptionGoogle = this.parentSubject.subscribe(  
    (coordenates: google.maps.LatLng) => {  
      this.setCoordenadas.emit(coordenates);  
      this.setearCoordenadas(coordenates);  
    }  
  );  
}  
  
// Nos desuscribimos para evitar perdidas de memoria  
ngOnDestroy(){  
  this.subscriptionGoogle.unsubscribe();  
}  
  
setearCoordenadas(coordenates: google.maps.LatLng) {  
  if (this.marker) {  
    this.marker.setMap(null); //Elimino los markers del mapa  
    google.maps.event.clearListeners(this.marker, 'dragend');  
  }  
  
  // Se agregan markers al mapa  
  if (this.map) {  
    this.map.setCenter(coordenadas);  
    this.marker = new google.maps.Marker({  
      map: this.map,  
      position: coordenadas,  
      draggable: true,  
    });  
    google.maps.event.addListener(this.marker, 'dragend', (evt) => {  
      this.setCoordenadas.emit(evt.latLng);  
    });  
  }  
}
```

```

ngAfterViewInit() {
  // Setea la posicion inicial del mapa
  if (!this.gmap) {
    return;
  }
  this.mapOptions.center = this.coordenates;
  this.map = new google.maps.Map(this.gmap.nativeElement, this.mapOptions);
  this.setearCoordenadas(this.coordenates);
}
}

```

### HU N°3 SignUp Patrocinador

Planilla de documentación

<b>Nombre de la HU</b>	Sign up Patrocinador
<b>Módulo seleccionado</b>	Seguridad
<b>Descripción breve de la HU</b>	Como potencial usuario con rol patrocinador quiero crear una cuenta para poder identificarme en el sistema.
<b>Actor</b>	Usuario
<b>Prioridad</b>	A
<b>Programador FrontEnd</b>	Ignacio Starkloff
<b>Programador BackEnd</b>	Juan Riera
<b>Tester</b>	Miguel Del Corso
<b>Aprobado por</b>	Elvio Aruta
<b>Denominación del programa</b>	<b>backend</b> "signup.controller.js" "signup.service.js" "signup.helper.js" <b>frontend</b> "SignUp.component.ts" "SignUp.component.html"
<b>Fecha inicio Programación</b>	09/08/2021

<b>Fecha finalización de programación</b>	13/08/2021
<b>Base de datos</b>	MongoDB Redis
<b>Colecciones involucradas</b>	Usuario Estado Usuario Estado Patrocinador Rol UsuarioRol
<b>Historias de usuarios relacionadas</b>	HU 6 - Login HU 1 - SignUp deportista HU 7 - Log Out
<b>Casos de prueba relacionados con resultados</b>	
<p>TCS-1 TCI-3            Crear usuario patrocinador con el campo cuit con el campo vacío=&gt; PASS            Crear usuario patrocinador con el campo cuit con Un cuit correcto=&gt; PASS            Crear usuario patrocinador con el campo cuit con un cuit con ceros=&gt; PASS            Crear usuario patrocinador con el campo cuit con un cuit con coma=&gt; PASS            Crear usuario patrocinador con el campo cuit con un cuit con punto=&gt; PASS            Crear usuario patrocinador con el campo cuit con un cuit negativo=&gt; PASS            Crear usuario patrocinador con el campo cuit con una expresión matemática=&gt; PASS            Crear usuario patrocinador con el campo cuit con una cadena de caracteres alfabéticos=&gt; PASS            Crear usuario patrocinador con el campo cuit con un carácter alfabético=&gt; PASS            Crear usuario patrocinador con el campo cuit con un carácter alfabético entre enteros=&gt; PASS            Crear usuario patrocinador con el campo cuit con un carácter especial=&gt; PASS            Crear usuario patrocinador con el campo cuit con null=&gt; PASS            Crear usuario patrocinador con el campo cuit con Nan=&gt; PASS            Crear usuario patrocinador con el campo cuit con NaN=&gt; PASS            Crear usuario patrocinador con el campo cuit con Na=&gt; PASS            Crear usuario patrocinador con el campo cuit con false=&gt; PASS            Crear usuario patrocinador con el campo cuit con true=&gt; PASS            Crear usuario patrocinador con el campo usuario con el campo vacío=&gt; PASS            Crear usuario patrocinador con el campo usuario con un usuario inexistente=&gt; PASS            Crear usuario patrocinador con el campo usuario con un usuario no habilitado=&gt; PASS            Crear usuario patrocinador con el campo usuario con un json mal formulado =&gt; PASS</p>	

## Documentación del BackEnd

Tiene la misma estructura que los anteriores, nada más que se le agrega el modelo del patrocinador:

```
const crearModeloPatrocinador = async (usuario) => {  
  // Datos que son necesarios para crear el modelo del patrocinador  
  const { CUIT, _id } = usuario;  
  // Creamos modelo del patrocinador  
  const modeloPatrocinador = new Patrocinador({  
    CUIT: CUIT,  
    usuario: _id,  
  });  
  // Guardamos al patrocinador en la base de datos  
  await savePatrocinadorService(modeloPatrocinador);  
  return modeloPatrocinador;  
};
```

## Pantallas



The screenshot shows the 'Sportinev' registration interface. At the top, the logo consists of two hands shaking in blue and green, followed by the text 'SPORTINEV'. Below the logo, the title 'Registrarse' is displayed. The form contains several input fields: 'Ingrese su nombre', 'Ingrese su teléfono o celular', 'Ingrese su email', 'Ingrese su contraseña' (with a toggle for visibility), 'Tipo de usuario' (a dropdown menu currently showing 'PATROCINADOR'), 'Ingrese su apellido', and 'Ingrese su CUIT'. At the bottom of the form, there are two buttons: a prominent blue 'Registrarse' button and a smaller, light blue 'Volver' button.

Figura 257: Pantalla de registro como patrocinador.

## Documentación del FrontEnd

Debido a que para la siguiente historia de usuario se usan los mismos componentes padres y services que fueron detallados anteriormente en SignUp Deportista, solo se detalla el componente hijo que es llamado una vez que se selecciona el tipo de rol patrocinador.

### Componente hijo patrocinador.component.html

```
<!-- Formulario específico para usuario tipo patrocinador -->
<form [formGroup]="form">
  <!-- Input para obtener apellido del nuevo usuario -->
  <div class="my-3">
    <mat-form-field appearance="legacy" style="width: 100%;">
      <mat-label>Ingrese su apellido</mat-label>
      <input matInput placeholder="Perez" formControlName="apellido"
autocomplete="off">
      <mat-error *ngIf="form.controls['apellido'].errors">El
        apellido es obligatorio</mat-error>
    </mat-form-field>
  </div>

  <!-- Input para obtener CUIT del nuevo usuario -->
  <div class="my-3">
    <mat-form-field appearance="legacy" style="width: 100%;">
      <mat-label>Ingrese su CUIT</mat-label>
      <input matInput placeholder="11222333" formControlName="CUIT"
autocomplete="off">
      <mat-error *ngIf="form.controls['CUIT'].errors">El
        CUIT es obligatorio</mat-error>
    </mat-form-field>
  </div>
</form>
```

### Componente hijo patrocinador.component.ts

```
import {
  Component,
  EventEmitter,
  Input,
  OnChanges,
  OnDestroy,
  OnInit,
  Output,
} from '@angular/core';
import { FormControl, FormGroup, Validators } from '@angular/forms';
import { blankSpaceValidator } from '@helpers/blankspaceValidator';
import { Subject, Subscription } from 'rxjs';

@Component({
  selector: 'app-patrocinador',
  templateUrl: './patrocinador.component.html',
  styleUrls: ['./patrocinador.component.scss'],
})
```

```
})
export class PatrocinadorComponent implements OnInit, OnDestroy, OnChanges {
  private subscriptionForm: Subscription = new Subscription();
  private subscriptionParentSubject: Subscription = new Subscription();
  hide: boolean = true;

  // Formulario especifico para usuarios de tipo patrocinador
  form: FormGroup = new FormGroup({
    apellido: new FormControl('', [Validators.required, blankSpaceValidator]),
    CUIT: new FormControl('', [Validators.required, blankSpaceValidator]),
  });

  // Datos obtenidos del componente padre
  @Input() parentSubject!: Subject<any>;
  @Input() usuarioViejo!: any;
  @Output() cambioDatos = new EventEmitter<any>();

  constructor() {}

  ngOnInit(): void {
    // Me suscribo a este evento para saber si el usuario tocó registrarse desde el
    // componente padre
    // y revisar que no haya errores en este componente hijo
    this.subscriptionParentSubject = this.parentSubject.subscribe((resp) => {
      if (resp) {
        this.form.markAllAsTouched();
      }
    });
    // LLamo a funciones para escuchar cambios en el formulario
    this.escucharCambiosEnForm();
  }

  // Este componente se reutiliza en módulo perfil, esto es para cargar datos anteriores
  ngOnChanges() {
    if (this.usuarioViejo != null) {
      this.cargarDatosAModificar();
    }
  }

  // Se cargan datos del usuario anteriores
  cargarDatosAModificar() {
    this.form.controls['apellido'].setValue(this.usuarioViejo.club.apellido);
    this.form.controls['CUIT'].setValue(this.usuarioViejo.club.CUIT);
  }

  // Cada vez que se produce un cambio en el formulario se envía la información al
  // componente padre
  escucharCambiosEnForm() {
    this.subscriptionForm = this.form.valueChanges.subscribe(() => {
      let data = this.form.getRawValue();
      let datos = {
        datos: {
          apellido: data.apellido,

```



```

        CUIT: data.CUIT,
      },
      estado: this.form.valid,
    };
    this.cambioDatos.emit(datos);
  });
}

// Me desuscribo de Las suscripciones para evitar pérdidas de memoria
ngOnDestroy(): void {
  this.subscriptionForm.unsubscribe();
  this.subscriptionParentSubject.unsubscribe();
}
}

```

## HU N°4 SignUp Organizador

### Planilla de documentación

<b>Nombre de la HU</b>	Sign up Organizador
<b>Módulo seleccionado</b>	Seguridad
<b>Descripción breve de la HU</b>	Como potencial usuario con rol organizador quiero crear una cuenta para poder identificarme en el sistema.
<b>Actor</b>	Usuario
<b>Prioridad</b>	A
<b>Programador BackEnd</b>	Ignacio Starkloff
<b>Programador FrontEnd</b>	Juan Riera
<b>Tester</b>	Elvio Aruta
<b>Aprobado por</b>	Jose Aranciba
<b>Denominación del programa</b>	<b>backend</b> "signup.controller.js" "signup.service.js" "signup.helper.js" <b>frontend</b> "SignUp.component.ts" "SignUp.component.html"
<b>Fecha inicio Programación</b>	09/08/2021

<b>Fecha finalización de programación</b>	13/08/2021
<b>Base de datos</b>	MongoDB Redis
<b>Colecciones involucradas</b>	Usuario Estado Usuario Estado Organizador Tipo Organizador Provincia Rol UsuarioRol
<b>Historias de usuarios relacionadas</b>	HU 6 - Login HU 1 - SignUp deportista HU 7 - Log Out
<b>Casos de prueba relacionados con resultados</b>	
<p>TCI-4</p> <p>Crear usuario organizador con el campo cuit con el campo vacío=&gt; PASS</p> <p>Crear usuario organizador con el campo cuit con Un cuit correcto=&gt; PASS</p> <p>Crear usuario organizador con el campo cuit con un cuit con ceros=&gt; PASS</p> <p>Crear usuario organizador con el campo cuit con un cuit con coma=&gt; PASS</p> <p>Crear usuario organizador con el campo cuit con un cuit con punto=&gt; PASS</p> <p>Crear usuario organizador con el campo cuit con un cuit negativo=&gt; PASS</p> <p>Crear usuario organizador con el campo cuit con una expresión matemática=&gt; PASS</p> <p>Crear usuario organizador con el campo cuit con una cadena de caracteres alfabéticos=&gt; PASS</p> <p>Crear usuario organizador con el campo cuit con un carácter alfabético=&gt; PASS</p> <p>Crear usuario organizador con el campo cuit con un carácter alfabético entre enteros=&gt; PASS</p> <p>Crear usuario organizador con el campo cuit con un carácter especial=&gt; PASS</p> <p>Crear usuario organizador con el campo cuit con null=&gt; PASS</p> <p>Crear usuario organizador con el campo cuit con Nan=&gt; PASS</p> <p>Crear usuario organizador con el campo cuit con NaN=&gt; PASS</p> <p>Crear usuario organizador con el campo cuit con Na=&gt; PASS</p> <p>Crear usuario organizador con el campo cuit con false=&gt; PASS</p> <p>Crear usuario organizador con el campo cuit con true=&gt; PASS</p> <p>Crear usuario organizador con el campo Tipo Organizador con el campo vacío=&gt; PASS</p> <p>Crear usuario organizador con el campo Tipo Organizador con un Tipo Organizador inexistente=&gt; PASS</p> <p>Crear usuario organizador con el campo Tipo Organizador con un Tipo Organizador no habilitado=&gt; PASS</p> <p>Crear usuario organizador con el campo Tipo Organizador con un json mal formulado =&gt; PASS</p> <p>Crear usuario organizador con el campo provincia con el campo vacío=&gt; PASS</p> <p>Crear usuario organizador con el campo provincia con un provincia inexistente=&gt; PASS</p> <p>Crear usuario organizador con el campo provincia con un provincia no habilitado=&gt; PASS</p> <p>Crear usuario organizador con el campo provincia con un json mal formulado =&gt; PASS</p>	



## Documentación del BackEnd

```
const crearModeloOrganizador = async (usuario) => {
  // Datos que son necesarios para crear el modelo del organizador
  const { tipoOrganizador, provincia, CUIT, _id } = usuario;
  if (tipoOrganizador && provincia && _id) {
    // Buscamos el tipoOrganizador en la base de datos
    const modeloTipoOrganizador = await findTipoOrganizadorByIdService(
      tipoOrganizador
    );
    // Buscamos la provincia en la base de datos
    const modeloProvincia = await findProvinciaByIdService(provincia);
    // Creamos modelo del organizador
    const modeloOrganizador = new Organizador({
      CUIT: CUIT,
      tipoOrganizador: modeloTipoOrganizador._id,
      provincia: modeloProvincia._id,
      usuario: _id,
    });

    // Guardamos al organizador en la base de datos
    await saveOrganizadorService(modeloOrganizador);
    return modeloOrganizador;
  } else {
    throw new BadRequest(
      "El formato enviado no es el adecuado para crear un organizador"
    );
  }
};
```

## Pantalla



The screenshot shows a registration form titled "Registrate" for the "SPORTINEV" system. The form is set against a dark background with light-colored text and input fields. The fields include:

- Ingrese su nombre
- Ingrese su teléfono o celular
- Ingrese su email
- Ingrese su contraseña (with a toggle for visibility)
- Tipo de usuario: ORGANIZADOR (dropdown menu)
- Ingrese su CUIT
- País (dropdown menu)
- Provincia (dropdown menu)

At the bottom of the form, there are two buttons: a prominent red "Registrate" button and a smaller, lighter "Volver" button.

Figura 258: Pantalla de registro como organizador.

## Documentación del FrontEnd

Debido a que para la siguiente historia de usuario se usan los mismos componentes padres y servicios que fueron detallados anteriormente en SignUp Deportista, solo se detalla el componente hijo que es llamado una vez que se selecciona el tipo de rol Organizador.

### Componente hijo organizador.component.html

```

<!-- Formulario específico para usuario tipo organizador -->
<form [formGroup]="form">
  <!-- Input para obtener CUIT del nuevo usuario -->
  <div class="my-3">
    <mat-form-field appearance="legacy" style="width: 100%;">
      <mat-label>Ingrese su CUIT</mat-label>
      <input matInput placeholder="20223293455" formControlName="CUIT"
        autocomplete="off">
  
```

```

        <mat-error *ngIf="form.controls['CUIT'].errors">El
            CUIT es obligatorio</mat-error>
        </mat-form-field>
    </div>

    <!-- Input para obtener pais del nuevo usuario -->
    <div class="my-3">
        <mat-form-field appearance="legacy" style="width: 100%;">
            <mat-label>País</mat-label>
            <mat-select formControlName="pais">
                <ng-container *ngFor="let pais of paises">
                    <mat-option [value]=pais>
                        {{pais.nombre}}
                    </mat-option>
                </ng-container>
            </mat-select>
            <mat-error *ngIf="form.controls['pais'].errors">El
                país es obligatorio</mat-error>
        </mat-form-field>
    </div>

    <!-- Input para obtener provincia del nuevo usuario se usa un filtro
        para mostrar solo las provincias del país seleccionado -->
    <div class="my-3">
        <mat-form-field appearance="legacy" style="width: 100%;">
            <mat-label>Provincia</mat-label>
            <mat-select formControlName="provincia">
                <ng-container *ngFor="let provincia of provincias |
filterProvinciaByPais:paisSeleccionado">
                    <mat-option [value]=provincia>
                        {{provincia.nombre}}
                    </mat-option>
                </ng-container>
            </mat-select>
            <mat-error *ngIf="form.controls['provincia'].errors">La provincia es
obligatoria</mat-error>
        </mat-form-field>
    </div>
</form>

```

### Componente hijo organizador.component.ts

```

import {
    Component,
    EventEmitter,
    Input,
    OnChanges,
    OnDestroy,
    OnInit,
    Output,
} from '@angular/core';
import { FormControl, FormGroup, Validators } from '@angular/forms';

```

```
import { blankSpaceValidator } from '@helpers/blankspaceValidator';
import { Subject, Subscription } from 'rxjs';

@Component({
  selector: 'app-organizador',
  templateUrl: './organizador.component.html',
  styleUrls: ['./organizador.component.scss'],
})
export class OrganizadorComponent implements OnInit, OnDestroy, OnChanges {
  private subscriptionForm: Subscription = new Subscription();
  private subscriptionPais: Subscription = new Subscription();
  private subscriptionProvincia: Subscription = new Subscription();
  private subscriptionParentSubject: Subscription = new Subscription();
  hide: boolean = true;

  // Formulario especifico para usuarios de tipo organizador
  form: FormGroup = new FormGroup({
    CUIT: new FormControl('', [
      Validators.required,
      blankSpaceValidator,
      Validators.pattern('^[0-9]*$'),
      Validators.minLength(8),
      Validators.maxLength(11),
    ]),
    provincia: new FormControl('', [Validators.required, blankSpaceValidator]),
    pais: new FormControl('', [Validators.required, blankSpaceValidator]),
  });

  // Datos obtenidos del componente padre
  @Input() paises: Array<any> = [];
  @Input() provincias: Array<any> = [];
  @Input() parentSubject!: Subject<any>;
  @Input() usuarioViejo!: any;
  @Output() cambioDatos = new EventEmitter<any>();

  paisSeleccionado: string = '';

  constructor() {}

  ngOnInit(): void {
    // Me suscribo a este evento para saber si el usuario toco registrarse desde el
    // componente padre
    // y revisar que no haya errores en este componente hijo
    this.subscriptionParentSubject = this.parentSubject.subscribe((resp) => {
      if (resp) {
        this.form.markAllAsTouched();
      }
    });
    // LLamo a funciones para escuchar cambios en el formulario
    this.escucharCambiosEnForm();
    this.cambiosEnPais();
  }
}
```

```
// Este componente se reutiliza en modulo perfil, esto es para cargar datos anteriores
ngOnChanges() {
  if (
    this.países.length > 0 &&
    this.provincias.length > 0 &&
    this.usuarioViejo != null
  ) {
    this.cargarDatosAModificar();
  }
}

// Se cargan datos del usuario anteriores
cargarDatosAModificar() {
  this.form.controls['CUIT'].setValue(this.usuarioViejo.club.CUIT);
  let provincia = this.provincias.find(
    (x) => x._id == this.usuarioViejo.organizador.provincia
  );
  let pais = this.países.find((x) => x._id == provincia?.pais);

  if (pais && provincia) {
    this.form.controls['pais'].setValue(pais);
    this.form.controls['provincia'].setValue(provincia);
  }
}

// Cada vez que se produce un cambio en el formulario se envia la información al
componente padre
escucharCambiosEnForm() {
  this.subscriptionForm = this.form.valueChanges.subscribe(() => {
    let data = this.form.getRawValue();

    let datos = {
      datos: {
        idProvincia: data.provincia._id,
        CUIT: data.CUIT,
      },
      estado: this.form.valid,
    };
    this.cambioDatos.emit(datos);
  });
}

// Cada vez que se cambia el pais, elimina la provincia y departamento seteados y se
actualiza el filtro
cambiosEnPais() {
  this.subscriptionPais = this.form.controls['pais'].valueChanges.subscribe(
    () => {
      this.paisSeleccionado = this.form.getRawValue().pais._id;
      this.form.controls['provincia'].setValue('');
    }
  );
}
```

```
// Me desuscribo de Las subscripciones para evitar perdidas de memoria
ngOnDestroy(): void {
  this.subscriptionForm.unsubscribe();
  this.subscriptionPais.unsubscribe();
  this.subscriptionProvincia.unsubscribe();
  this.subscriptionParentSubject.unsubscribe();
}
}
```

## HU N°5 SignUp Árbitro

### Planilla de documentación

<b>Nombre de la HU</b>	Sign up Árbitro
<b>Módulo seleccionado</b>	Seguridad
<b>Descripción breve de la HU</b>	Como potencial usuario con rol árbitro quiero crear una cuenta para poder identificarme en el sistema.
<b>Actor</b>	Usuario
<b>Prioridad</b>	A
<b>Programador FrontEnd</b>	Juan Segundo Riera
<b>Programador BackEnd</b>	Ignacio Starkloff
<b>Tester</b>	Miguel Del Corso
<b>Aprobado por</b>	Elvio Aruta
<b>Denominación del programa</b>	<b>backend</b> "signup.controller.js" "signup.service.js" "signup.helper.js" <b>frontend</b> "SignUp.component.ts" "SignUp.component.html"
<b>Fecha inicio Programación</b>	09/08/2021
<b>Fecha finalización de programación</b>	13/08/2021
<b>Base de datos</b>	MongoDB Redis
<b>Colecciones involucradas</b>	Usuario Estado




	Usuario Estado Arbitro Provincia Rol UsuarioRol
<b>Historias de usuarios relacionadas</b>	HU 6 - Login HU 1 - SignUp deportista HU 7 - Log Out
<b>Casos de prueba relacionados con resultados</b>	
<p>TCS-3</p> <p>Crear usuario árbitro con el campo dni con el campo vacío=&gt; PASS</p> <p>Crear usuario árbitro con el campo dni con Un número correcto=&gt; PASS</p> <p>Crear usuario árbitro con el campo dni con el número cero=&gt; PASS</p> <p>Crear usuario árbitro con el campo dni con un número con coma=&gt; PASS</p> <p>Crear usuario árbitro con el campo dni con un número con punto=&gt; PASS</p> <p>Crear usuario árbitro con el campo dni con un número negativo=&gt; PASS</p> <p>Crear usuario árbitro con el campo dni con una expresión matemática=&gt; PASS</p> <p>Crear usuario árbitro con el campo dni con una cadena de caracteres alfabéticos=&gt; PASS</p> <p>Crear usuario árbitro con el campo dni con un carácter alfabético=&gt; PASS</p> <p>Crear usuario árbitro con el campo dni con un carácter alfabético entre enteros=&gt; PASS</p> <p>Crear usuario árbitro con el campo dni con un carácter especial=&gt; PASS</p> <p>Crear usuario árbitro con el campo dni con null=&gt; PASS</p> <p>Crear usuario árbitro con el campo dni con Nan=&gt; PASS</p> <p>Crear usuario árbitro con el campo dni con NaN=&gt; PASS</p> <p>Crear usuario árbitro con el campo dni con Na=&gt; PASS</p> <p>Crear usuario árbitro con el campo dni con false=&gt; PASS</p> <p>Crear usuario árbitro con el campo dni con true=&gt; PASS</p> <p>Crear usuario árbitro con el campo Deporte con el campo vacío=&gt; PASS</p> <p>Crear usuario árbitro con el campo Deporte con un Deporte inexistente=&gt; PASS</p> <p>Crear usuario árbitro con el campo Deporte con un Deporte no habilitado=&gt; PASS</p> <p>Crear usuario árbitro con el campo Deporte con un json mal formulado =&gt; PASS</p> <p>Crear usuario árbitro con el campo provincia con el campo vacío=&gt; PASS</p> <p>Crear usuario árbitro con el campo provincia con un provincia inexistente=&gt; PASS</p> <p>Crear usuario árbitro con el campo provincia con un provincia no habilitado=&gt; PASS</p> <p>Crear usuario árbitro con el campo provincia con un json mal formulado =&gt; PASS</p> <p>Crear usuario árbitro con el campo usuario con el campo vacío=&gt; PASS</p> <p>Crear usuario árbitro con el campo usuario con un usuario inexistente=&gt; PASS</p> <p>Crear usuario árbitro con el campo usuario con un usuario no habilitado=&gt; PASS</p> <p>Crear usuario árbitro con el campo usuario con un json mal formulado =&gt; PASS</p>	

## Documentación del BackEnd

```
const crearModeloArbitro = async (usuario) => {
  // Datos que son necesarios para crear el modelo del arbitro
  const { DNI, provincia, deporte, _id } = usuario;
```

```
if (provincia && deporte && _id) {  
  // Buscamos el deporte en la base de datos  
  const modeloDeporte = await findDeporteByIdService(deporte);  
  // Buscamos la provincia en la base de datos  
  const modeloProvincia = await findProvinciaByIdService(provincia);  
  // Creamos modelo del Árbitro  
  const modeloArbitro = new Arbitro({  
    DNI: DNI,  
    deporte: modeloDeporte._id,  
    provincia: modeloProvincia._id,  
    usuario: _id,  
  });  
  // Guardamos al árbitro en la base de datos  
  await saveArbitroService(modeloArbitro);  
  return modeloArbitro;  
} else {  
  throw new BadRequest(  
    "El formato enviado no es el adecuado para crear un árbitro"  
  );  
}  
};
```

## Pantallas



The screenshot shows a registration form for 'SPORTINEV'. The form is titled 'Registrarse' and includes the following fields and elements:

- Input field: 'Ingrese su nombre'
- Input field: 'Ingrese su teléfono o celular'
- Input field: 'Ingrese su email'
- Input field: 'Ingrese su contraseña' with a toggle icon for visibility.
- Dropdown menu: 'Tipo de usuario' with 'ARBITRO' selected.
- Input field: 'Ingrese su apellido'
- Input field: 'Ingrese su DNI'
- Dropdown menu: 'Pais'
- Dropdown menu: 'Provincia'
- Buttons: 'Registrarse' (highlighted in red) and 'Volver'.

Figura 259: Pantalla de registro como árbitro.

## Documentación del FrontEnd

Debido a que para la siguiente historia de usuario se usan los mismos componentes padres y services que fueron detallados anteriormente en SignUp Deportista, solo se detalla el componente hijo que es llamado una vez que se selecciona el tipo de rol Árbitro.

### Componente hijo arbitro.component.html

```
<!-- Formulario específico para usuario tipo arbitro -->  
<form [formGroup]="form">  
  <!-- Input para obtener apellido del nuevo usuario -->  
  <div class="my-3">
```

```
<mat-form-field appearance="legacy" style="width: 100%;">
  <mat-label>Ingrese su apellido</mat-label>
  <input matInput placeholder="Perez" formControlName="apellido"
autocomplete="off">
  <mat-error *ngIf="form.controls['apellido'].errors">El
    apellido es obligatorio</mat-error>
</mat-form-field>
</div>

<!-- Input para obtener DNI del nuevo usuario -->
<div class="my-3">
  <mat-form-field appearance="legacy" style="width: 100%;">
    <mat-label>Ingrese su DNI</mat-label>
    <input matInput placeholder="1122333" formControlName="DNI"
autocomplete="off">
    <mat-error *ngIf="form.controls['DNI'].errors">El
      DNI es obligatorio</mat-error>
  </mat-form-field>
</div>

<!-- Input para obtener pais del nuevo usuario -->
<div class="my-3">
  <mat-form-field appearance="legacy" style="width: 100%;">
    <mat-label>País</mat-label>
    <mat-select formControlName="pais">
      <ng-container *ngFor="let pais of paises">
        <mat-option [value]=pais>
          {{pais.nombre}}
        </mat-option>
      </ng-container>
    </mat-select>
    <mat-error *ngIf="form.controls['pais'].errors">El
      país es obligatorio</mat-error>
  </mat-form-field>
</div>

<!-- Input para obtener provincia del nuevo usuario se usa un filtro
para mostrar solo las provincias del pais seleccionado -->
<div class="my-3">
  <mat-form-field appearance="legacy" style="width: 100%;">
    <mat-label>Provincia</mat-label>
    <mat-select formControlName="provincia">
      <ng-container *ngFor="let provincia of provincias |
filterProvinciaByPais:paisSeleccionado">
        <mat-option [value]=provincia>
          {{provincia.nombre}}
        </mat-option>
      </ng-container>
    </mat-select>
    <mat-error *ngIf="form.controls['provincia'].errors">La provincia es
obligatoria</mat-error>
  </mat-form-field>
</div>
```

```
</form>
```

## Componente hijo arbitro.component.ts

```
import {
  Component,
  EventEmitter,
  Input,
  OnChanges,
  OnDestroy,
  OnInit,
  Output,
} from '@angular/core';
import { FormControl, FormGroup, Validators } from '@angular/forms';
import { blankSpaceValidator } from '@helpers/blankspaceValidator';
import { Subject, Subscription } from 'rxjs';

@Component({
  selector: 'app-arbitro',
  templateUrl: './arbitro.component.html',
  styleUrls: ['./arbitro.component.scss'],
})
export class ArbitroComponent implements OnInit, OnDestroy, OnChanges {
  private subscriptionForm: Subscription = new Subscription();
  private subscriptionPais: Subscription = new Subscription();
  private subscriptionParentSubject: Subscription = new Subscription();
  hide: boolean = true;

  // Formulario especifico para usuarios de tipo arbitro
  form: FormGroup = new FormGroup({
    apellido: new FormControl('', [Validators.required, blankSpaceValidator]),
    DNI: new FormControl('', [Validators.required, blankSpaceValidator]),
    pais: new FormControl('', [Validators.required, blankSpaceValidator]),
    provincia: new FormControl('', [Validators.required, blankSpaceValidator]),
    deporte: new FormControl('', [Validators.required, blankSpaceValidator]),
  });

  // Datos obtenidos del componente padre
  @Input() paises: Array<any> = [];
  @Input() provincias: Array<any> = [];
  @Input() parentSubject!: Subject<any>;
  @Input() usuarioViejo!: any;
  @Output() cambioDatos = new EventEmitter<any>();

  paisSeleccionado: string = '';

  constructor() {}

  ngOnInit(): void {
    // Me suscribo a este evento para saber si el usuario toco registrarse desde el
    // componente padre
    // y revisar que no haya errores en este componente hijo
  }
}
```

```

this.subscriptionParentSubject = this.parentSubject.subscribe((resp) => {
  if (resp) {
    this.form.markAllAsTouched();
  }
});
// LLamo a funciones para escuchar cambios en el formulario
this.escucharCambiosEnForm();
this.cambiosEnPais();
}

// Este componente se reutiliza en modulo perfil, esto es para cargar datos anteriores
ngOnChanges() {
  if (
    this.paises.length > 0 &&
    this.provincias.length > 0 &&
    this.usuarioViejo != null
  ) {
    this.cargarDatosAModificar();
  }
}

// Se cargan datos del usuario anteriores
cargarDatosAModificar() {
  this.form.controls['apellido'].setValue(this.usuarioViejo.arbitro.apellido);
  this.form.controls['DNI'].setValue(this.usuarioViejo.arbitro.DNI);

  let provincia = this.provincias.find(
    (x) => x._id == this.usuarioViejo.arbitro.provincia
  );
  let pais = this.paises.find((x) => x._id == provincia?.pais);

  if (pais && provincia) {
    this.form.controls['pais'].setValue(pais);
    this.form.controls['provincia'].setValue(provincia);
  }
}

// Cada vez que se produce un cambio en el formulario se envia la información al
componente padre
escucharCambiosEnForm() {
  this.subscriptionForm = this.form.valueChanges.subscribe(() => {
    let data = this.form.getRawValue();
    console.log(data);

    let datos = {
      datos: {
        apellido: data.apellido,
        idProvincia: data.provincia._id,
        DNI: data.DNI,
        idDeporte: data.deporte,
      },
      estado: this.form.valid,
    };
  });
}

```

```

        this.cambioDatos.emit(datos);
    });
}

// Cada vez que se cambia el pais, elimina la provincia y departamento seteados y se
actualiza el filtro
cambiosEnPais() {
    this.subscriptionPais = this.form.controls['pais'].valueChanges.subscribe(
        () => {
            this.paisSeleccionado = this.form.getRawValue().pais._id;
            this.form.controls['provincia'].setValue('');
        }
    );
}

// Me desuscribo de las subscripciones para evitar perdidas de memoria
ngOnDestroy(): void {
    this.subscriptionForm.unsubscribe();
    this.subscriptionPais.unsubscribe();
    this.subscriptionParentSubject.unsubscribe();
}
}

```

HU N°6 LogIn

Planilla de documentación

<b>Nombre de la HU</b>	LogIn
<b>Módulo seleccionado</b>	Seguridad
<b>Descripción breve de la HU</b>	Como usuario quiero identificarme en el sistema con mi email y contraseña para poder hacer uso de las funcionalidades del sistema.
<b>Actor</b>	Usuario
<b>Prioridad</b>	A
<b>Programador BackEnd</b>	Juan Segundo Riera
<b>Programador FrontEnd</b>	Ignacio Starkloff
<b>Tester</b>	Jose Aranciba
<b>Aprobado por</b>	Elvio Aruta
<b>Denominación del programa</b>	<b>backend</b>

	"login.controller.js" "login.service.js" "login.helper.js" <b>frontend</b> "login.component.ts" "login.component.html"
<b>Fecha inicio Programación</b>	09/08/2021
<b>Fecha finalización de programación</b>	13/08/2021
<b>Base de datos</b>	MongoDB Redis
<b>Colecciones involucradas</b>	Usuario Estado Usuario Estado Organizador Administrador Árbitro Patrocinador Club Rol UsuarioRol
<b>Historias de usuarios relacionadas</b>	HU 7 - Log Out
<b>Casos de prueba relacionados con resultados</b>	
TCU-01 TCU-02 TCU-03 TCS-3 TCS-1 TCS-4 TCI-1 TCI-2 TCI-3 TCI-4 TCI-5 TCC-1 TCC-2 TCC-3 TCE-1 TCC-2 TCC-3 Loguear un usuario con el campo correo con un solo mail vacío=> PASS Loguear un usuario con el campo correo con un mail ya asociado a un usuario=> PASS Loguear un usuario con el campo correo con Un mail sin caracteres previos al arroba => PASS Loguear un usuario con el campo correo con el carácter: " => PASS Loguear un usuario con el campo correo con Un mail con más de un punto luego de el	



arroba => PASS  
Loguear un usuario con el campo correo con Un mail terminando en punto => PASS  
Loguear un usuario con el campo correo con Un mail sin puntos luego de el arroba => PASS  
Loguear un usuario con el campo correo con Un mail con caracteres especiales=> PASS  
Loguear un usuario con el campo correo con Un mail con más de un @=> PASS  
Loguear un usuario con el campo correo con Un mail con el carácter: -=> PASS  
Loguear un usuario con el campo correo con Un mail con el carácter: \_=> PASS  
Loguear un usuario con el campo correo con Un mail con menos de 2 caracteres luego del punto final=> PASS  
Loguear un usuario con el campo correo con Un mail con menos de 1 caracter luego del arroba y previo al punto=> PASS  
Loguear un usuario con el campo correo con Un mail con mas de 11 caracteres=> PASS  
Loguear un usuario con el campo contraseña con una contraseña vacío=> PASS  
Loguear un usuario con el campo contraseña con una contraseña incorrecta=> PASS  
Loguear un usuario con el campo contraseña con una contraseña correcta=> PASS  
Loguear un usuario con el campo contraseña con una contraseña anterior=> PASS  
Loguear un usuario con el campo contraseña con un json mal formulado => PASS  
Loguear un usuario con el campo contraseña con los caracteres: \" => PASS  
Loguear un usuario con el campo contraseña con Dos caracteres: \ => PASS  
Loguear un usuario con el campo rol con el campo vacío=> PASS  
Loguear un usuario con el campo rol con un rol inexistente=> PASS  
Loguear un usuario con el campo rol con un rol no habilitado=> PASS  
Loguear un usuario con el campo rol con un json mal formulado => PASS

## Documentación del BackEnd

Identifica en el sistema al usuario con un correo, contraseña y rol específico.

**URL:** \${URL}/api/auth/signIn

**Method:** POST

### BODY

```
{  
  "correo": "vofagi2535@gmail.com",  
  "password": "789456",  
  "rol": "DEPORTISTA" //El rol varia  
}
```

### RESPONSE

```
{  
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9",  
  "usuario": {
```

```

"fechaInicioVig": "2021-08-19T14:06:36.647Z",
"_id": "611e6570f3139924224264c4",
"nombre": "Miguel",
"correo": "vofagi2535@kembung.com",
"estados": [
  {
    "fechaInicioVig": "2021-08-19T14:06:36.646Z",
    "_id": "611e6570f3139924224264c6",
    "estado": "6115fc6c8fd6a830f52b2c8f"
  }
],
"roles": [
  {
    "fechaInicioVig": "2021-08-19T14:06:36.646Z",
    "_id": "611e6570f3139924224264c5",
    "rol": "6113601985887dfb2d950257"
  }
],
"deportista": {
  "fechaInicioVig": "2021-08-19T14:06:36.673Z",
  "_id": "611e6570f3139924224264c9",
  "apellido": "Miguel Del Corso",
  "usuario": "611e6570f3139924224264c4",
  "deportes": [],
  "__v": 0
}
},
"rol": "DEPORTISTA"
}

```

### auth.routes.js

```

router.post(
  "/signin",
  [
    check("correo", "El correo ingresado no es válido").isEmail(),
    check("password", "La contraseña no es válida").not().isEmpty(),
    validarCampos,
  ],
  asyncHandler(signin)
);

```

### auth.controller.js

```

const signin = async (req = request, res = response) => {
  const { correo, password, rol } = req.body;

  // Buscamos al usuario en la base de datos
  let usuario = await findUsuarioActivoByEmailService(correo);
  // Si el usuario no fue encontrado o la contraseña es incorrecta

```

```
if (!usuario || !Usuario.comparePassword(password, usuario.password)) {
  throw new NotFound("El email o la contraseña son incorrectos");
}
// Obtenemos Los nombre de Los roles que no estén con fechaFinVig
const rolesUsuario = await obtenerRolesActivos(usuario.roles);
// Si el no esta entre La lista de roles en Los cuales el usuario esta activo
if (!rolesUsuario.includes(rol)) {
  throw new NotPermission("El usuario no cuenta con el rol indicado");
}
usuario = await populateUsuarioRol(usuario, rol);

// Generar JWT
const token = await generarJWT(usuario.correo, rol);
let permisos = rolesUsuario
  .filter((rolu) => rolu.nombre.toUpperCase() === rol.toUpperCase())
  .map((rolu) => {
    return rolu.rolPermisos;
  });

return res.json({
  token,
  usuario,
  rol
});
};
```

## Método generarJWT

```
// Generar JWT
const token = await generarJWT(usuario.correo, rol);
const publicidad = await findPublicidadActivosService();
return res.json({
  token,
  usuario,
  permisos,
  rol,
  publicidad,
});
};
```

## Roles Especiales

En caso de que un usuario quiera loguearse con un rol especial, esta funcionalidad es manejada por el controlador de roles especiales: "rolcontroller"

Una de las funciones mas importantes de el codigo, son obtener los roles activos, los roles básicos y los roles especiales. Se puede observar que se hace una diferenciación entre los

roles especiales y los roles básicos predeterminados del sistema, ya que estos roles predeterminados cargan con la lógica básica inicial y necesaria de todo el sistema y un cambio en estos significaron un cambio en la funcionalidad general del sistema y requiere un análisis funcional más exhaustivo que los roles especiales, que son aquellos que serán parametrizables por parte de un administrador con permisos para crearlos modificarlos y borrarlos.

```
const getRoles = async (
  { query: { nombre, limit = 10, page = 1, sort = "asc" } },
  res
) => {
  const rolesBasicos = [
    "DEPORTISTA",
    "ORGANIZADOR",
    "CLUB",
    "ARBITRO",
    "PATROCINADOR",
    "ADMINISTRADOR",
  ];
  const roles = nombre
    ? await findRolesfilltradosService(nombre, limit, page, sort)
    : await findRolesService(limit, page, sort);

  roles.docs = roles.docs.filter((rol) => {
    return !rolesBasicos.includes(rol.nombre);
  });
  res.json({
    data: roles,
  });
};

const getRolesBasicosConId = async (req, res) => {
  const rolesBasicos = [
    "DEPORTISTA",
    "ORGANIZADOR",
    "CLUB",
    "ARBITRO",
    "PATROCINADOR",
  ];
  let roles = await findRolesService(99999);
  let rolesBasicosConId = roles.docs.filter((rol) => {
    return rolesBasicos.includes(rol.nombre);
  });
  res.json({
    rolesBasicos: rolesBasicosConId,
  });
};

const getRolesActivosUsuario = async (req, res) => {
  const { usuarioAutenticado } = req;
  let idRolesActivosUsuario = usuarioAutenticado.roles.map((rol) => {
    if (!rol.fechaFinVig) {
      return rol.rol;
    }
  });
};
```

```
    }  
  });  
  const rolesBasicos = [  
    "DEPORTISTA",  
    "ORGANIZADOR",  
    "CLUB",  
    "ARBITRO",  
    "PATROCINADOR",  
  ];  
  let roles = await findRolesService(99999);  
  let rolesBasicosConId = roles.docs.filter((rol) => {  
    return rolesBasicos.includes(rol.nombre);  
  });  
  
  let rolesUsuarioActivos = rolesBasicosConId.filter((rol) => {  
    if (idRolesActivosUsuario.includes(rol._id)) {  
      return rol;  
    }  
  });  
  
  res.json({  
    rolesBasicos: rolesUsuarioActivos,  
  });  
};
```

Estas funciones son necesarias para principalmente, la carga de los roles al momento de loguear al usuario.

Por último, las funciones principales de este controlador son aquellas que asignan y revocan los roles y, por lo tanto, los permisos especiales a los usuarios del sistema.

```
const asignarRolEspecialUsuario = async (req, res) => {  
  const { rol } = req;  
  const { idsUsuarios, idRol } = req.body;  
  await controlarAccionService({  
    rolUsuario: rol,  
    rolesPermitidos: ["ADMINISTRADOR"],  
  });  
  
  await Promise.all(  
    idsUsuarios.map(async (idUsuario) => {  
      if (ObjectId.isValid(idUsuario)) {  
        await asignarRolUsuarioService(idUsuario, idRol);  
      }  
    })  
  );  
  res.json("Se asignó roles con éxito");  
};
```

```
const desasignarRolEspecialUsuario = async (req, res) => {  
  const { rol } = req;  
  await controlarAccionService({  
    rolUsuario: rol,  
    rolesPermitidos: ["ADMINISTRADOR"],  
  });  
  const { idUsuario, idsRoles } = req.body;  
  await Promise.all(  
    idsRoles.map(async (idRol) => {  
      if (ObjectId.isValid(idRol)) {  
        await desasignarRolUsuarioService(idUsuario, idRol);  
      }  
    })  
  );  
  res.json("Roles desasignados con éxito");  
};
```

## Pantallas

**SPORTINEU**

### Iniciar sesión

Tipo de usuario

Ingrese su email

Ingrese su contraseña

[¿Olvidaste tu contraseña?](#)

**Ingresar**

Volver

[¿No sos miembro?](#)

Registrate

Figura 260: Pantalla de inicio de sesión.

## Documentación del FrontEnd

### Componente hijo login.component.html

```

<div class="container mt-5">
  <div class="row">
    <!-- Logo de la aplicación -->
    <div class="col-12 text-center">
      
    </div>
  </div>
  <!-- Componente card nos da una estructura responsive solo hay que pasarle el form -->
  <app-card-form [componentePasado]=loginForm titulo="Iniciar sesión"></app-card-form>
</div>

<!-- Form que se le enviara a card -->
<ng-template #loginForm>
  <form [formGroup]="form" (ngSubmit)="enviarInformacion()">

    <!-- Input para obtener tipo de rol del usuario -->
    <div class="my-3">
      <mat-form-field appearance="legacy" style="width: 100%;">
        <mat-label>Tipo de usuario</mat-label>
        <mat-select formControlName="rol">
          <ng-container *ngFor="let rol of roles">
            <mat-option [value]=rol>
              {{rol}}
            </mat-option>
          </ng-container>
        </mat-select>
      </mat-form-field>
    </div>

    <!-- Input para obtener email del usuario -->
    <div class="my-3">
      <mat-form-field appearance="legacy" style="width: 100%;">
        <mat-label>Ingrese su email</mat-label>
        <input matInput placeholder="pat@example.com" formControlName="email"
autocomplete="off">
        <mat-error *ngIf="form.controls['email'].errors">El
          email ingresado no es válido</mat-error>
      </mat-form-field>
    </div>

    <!-- Input para obtener contraseña del usuario -->
    <div>
      <mat-form-field appearance="legacy" style="width: 100%;">
        <mat-label>Ingrese su contraseña</mat-label>
        <input matInput [type]="hide ? 'password' : 'text'" placeholder="123456"
formControlName="password"
autocomplete="off">
        <button mat-icon-button matSuffix (click)="hide = !hide" [attr.aria-label]='Hide
password' ">

```

```

    [attr.aria-pressed]="hide" type="button">
    <mat-icon>{{hide ? 'visibility_off' : 'visibility'}}</mat-icon>
  </button>
  <mat-error *ngIf="form.controls['password'].errors">La contraseña debe tener al
menos 6
    caracteres</mat-error>
  </mat-form-field>
</div>

<!-- Link para abrir modal para recuperar contraseña -->
<p class="text-right mb-4">
  <a (click)="olvidoContrasenna()" class="cursor">¿Olvidaste tu contraseña?</a>
</p>

<!-- Botones para logearse o para volver a la landing page -->
<div class="row mt-3">
  <div class="col-12 text-center mb-2">
    <button mat-raised-button color="primary" style="width: 100%;">Ingresar</button>
  </div>
  <div class="col-12 text-center mb-2">
    <button mat-stroked-button color="primary" style="width: 100%;
(click)="volver()">Volver</button>
  </div>
</div>

<!-- Redireccionar a usuario a pagina de signup -->
<h5 class="text-center mt-4">¿No sos miembro?</h5>
<div class="row mt-3">
  <div class="col-12 text-center">
    <button mat-raised-button class="btn-dark" style="width: 60%;
(click)="irASignup()">Registrate</button>
  </div>
</div>
</form>
</ng-template>

```

### Componente hijo login.component.ts

```

import { Component, OnInit } from '@angular/core';
import {
  FormBuilder,
  FormControl,
  FormGroup,
  Validators,
} from '@angular/forms';
import { Router } from '@angular/router';
import { blankSpaceValidator } from '@helpers/blankspaceValidator';
import { Store } from '@ngrx/store';
import { DialogsService } from '@services/dialogs.service';
import { AppState } from '@store/app.reducers';
import { AuthService } from '../services/auth.service';
import { GlobalService } from '../services/global.service';
import { setUsuarioActual } from '../store/actions/configuracion.action';

```



```
@Component({
  selector: 'app-login',
  templateUrl: './login.component.html',
  styleUrls: ['./login.component.scss'],
})
export class LoginComponent implements OnInit {
  hide: boolean = true;

  // Form para obtener datos necesario para Login
  form: FormGroup = new FormGroup({
    email: new FormControl('', [
      Validators.required,
      Validators.email,
      blankSpaceValidator,
    ]),
    password: new FormControl('', [Validators.required, blankSpaceValidator]),
    rol: new FormControl('', [Validators.required, blankSpaceValidator]),
  });

  // Los roles que puede seleccionar, se obtienen conectandose con el backend
  roles: Array<any> = [];

  constructor(
    private router: Router,
    public FormBuilder: FormBuilder,
    private dialogService: DialogService,
    private authService: AuthService,
    private globalService: GlobalService,
    private store: Store<AppState>
  ) {}

  ngOnInit(): void {
    this.getData(); //Buscar data para Login
  }

  // Busca los roles que puede seleccionar el usuario a la hora de logearse
  getData() {
    this.globalService.getRolesBasicos().subscribe(
      (respuesta) => {
        // Se setean los roles
        this.roles = respuesta.rolesBasicos;
      },
      (error) => {
        // En caso de error se le informa al usuario y se lo redirecciona a la landing
        // page
        this.dialogService.cerrar();
        this.dialogService.toastError(
          'Se produjo un error, vuelva a intentarlo más tarde'
        );
        this.router.navigateByUrl('inicio');
      }
    );
  }
};
```

```
}

// Logearse en el sistem
enviarInformacion() {
  if (this.form.invalid) {
    return;
  }
  //enviar informacion
  this.dialogsService.cargando('Iniciando sesión');
  let datosUsuario = this.form.getRawValue();
  let data = {
    correo: datosUsuario.email,
    password: datosUsuario.password,
    rol: datosUsuario.rol,
  };

  // Se llama a un servicio que se conectara con una api que nos permite Logearnos
  this.authService.login(data).subscribe(
    (respuesta) => {
      // En caso de que nos logeamos con exito se guarda el usuario logeado en redux
      // Y despues esta información se puede acceder desde toda la aplicación
      this.store.dispatch(
        new setUsuarioActual({
          usuario: respuesta.usuario,
          token: respuesta.token,
          rol: respuesta.rol,
        })
      );
      this.dialogsService.cerrar();
      this.router.navigateByUrl('home');
    },
    (error) => {
      // En caso de error se le informa del mismo al usuario
      this.dialogsService.cerrar();
      this.dialogsService.toastError(
        'El email o la contraseña son incorrectos o no han sido verificados'
      );
    }
  );
}

// Se vuelve a la Landing page
volver() {
  this.router.navigate(['inicio']);
}

// Función que se llama desde el html para ir al signup
irASignup() {
  this.router.navigate(['signup']);
}

// Función que llama a un service que llama al modal para recuperar contraseña
olvidoContrasenna() {
```



<b>Actor</b>	Usuario
<b>Prioridad</b>	A
<b>Programador BackEnd</b>	Ignacio Starkloff
<b>Programador FrontEnd</b>	Juan Riera
<b>Tester</b>	Elvio Aruta
<b>Aprobado por</b>	Miguel Del Corso
<b>Denominación del programa</b>	<b>backend</b> "auth.controller.js" "auth.service.js" <b>frontend</b> "navbar.component.ts" "navbar.component.html"
<b>Fecha inicio Programación</b>	09/08/2021
<b>Fecha finalización de programación</b>	13/08/2021
<b>Base de datos</b>	MongoDB Redis
<b>Colecciones involucradas</b>	Usuario
<b>Historias de usuarios relacionadas</b>	HU 6 - Login
<b>Casos de prueba relacionados con resultados</b>	
Log out exitoso => PASS Log out presionando repetidamente el botón de log out => PASS	

#### Documentación del BackEnd

Cierra la sesión de un usuario.

- URL: {URL}/api/auth/logout
- Method: POST

#### HEADER

Authorization = \${token}

#### Response

Ignacio Starkloff, Juan Segundo Riera,  
Miguel Del Corso, Jose Aranciba,  
Elvio Aruta.

```
{  
  "message": "La sesión se ha cerrado con éxito"  
}
```

### auth.routes.js

```
router.post("/logout", asyncHandler(logout));
```

### auth.controller.js

```
const logout = async (req = request, res = response) => {  
  const idToken = req.header("Authorization");  
  // Guardamos el JWT en la lista negra de redis  
  await logoutService(idToken);  
  
  res.json({  
    message: "La sesión se ha cerrado con éxito",  
  });  
};
```

### auth.service.js

```
const logoutService = async (token) => {  
  try {  
    // Guardamos el documento en redis  
    await client.hmset(`token#${token}`, "body", `${token}`);  
    // Seteamos 4hs para que expire el token y se elimine solo  
    await client.expire(`token#${token}`, 60 * 60 * 4);  
    return token;  
  } catch (error) {  
    console.log(error);  
    throw new GeneralError("Se produjo un error a la hora de guardar el token");  
  }  
};
```

## Pantallas

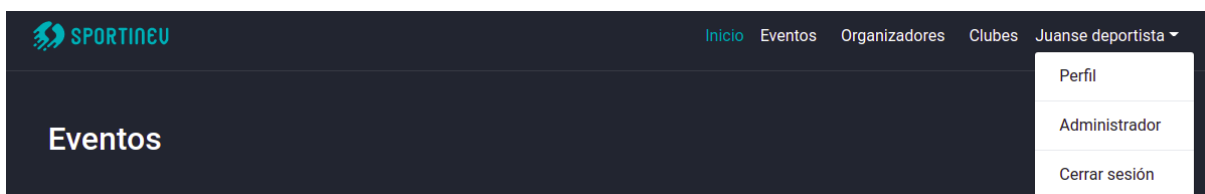


Figura 261: Pantalla que nos muestra la solapa de cerrar sesión.

Documentación del FrontEnd

### Componente navbar.component.html

```

<!-- Barra de navegación de usuarios logeados -->
<nav class="navbar navbar-expand-lg navbar-dark bg-dark example-toolbar mb-5 ">
  <div class="container container-fluid px-0">
    <a class="navbar-brand" href="/#/inicio" style="color: white;">
      
    </a>
    <button class="navbar-toggler" type="button" data-toggle="collapse"
    data-target="#navbarNav"
      aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <!-- Links para redireccionar a otra pagina -->

      <ul class="navbar-nav ml-auto">
        <li class="nav-item itemPersonalizado">
          <a class="nav-link routerLinkActive="active-link"
            [routerLink]="['/home']">Inicio</a>
        </li>
        <li class="nav-item itemPersonalizado mr-2">
          <a class="nav-link routerLinkActive="active-link"
            [routerLink]="['/eventos']">Eventos</a>
        </li>
        <li class="nav-item itemPersonalizado mr-2">
          <a class="nav-link routerLinkActive="active-link"
            [routerLink]="['/organizadores']">Organizadores</a>
        </li>
        <li class="nav-item itemPersonalizado mr-2">
          <a class="nav-link routerLinkActive="active-link"
            [routerLink]="['/clubes']">Clubes</a>
        </li>

        <!-- Menu dropdown que se despliega al presionar el nombre del usuario -->
        <div class="dropdown" id="dropdown">
          <a class="btn btnIconosTransparente dropdown-toggle p-0" type="button"
            id="dropdownMenuButton"
              data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
            <span class="fas fa-users"
              aria-hidden="true"></span><span>{{nombreUsuario}}</span>
            </a>

          <!-- Opciones del menu dropdwon -->
          <div class="dropdown-menu" aria-labelledby="navbarDropdown">
            <a class="dropdown-item" [routerLink]="['/perfil']">Perfil</a>

```

```

        <div class="dropdown-divider"></div>
        <a class="dropdown-item" [routerLink]="['/administrador']">Administrador</a>
        <div class="dropdown-divider"></div>
        <!-- Boton para desloguearse del sistema -->
        <a class="dropdown-item" (click)="salir()">Cerrar sesión</a>
    </div>
</div>
</ul>
</div>
</div>
</nav>

```

### Componente navbar.component.ts

```

import { Component, OnInit, OnDestroy } from '@angular/core';
import { Router } from '@angular/router';
import { Store } from '@ngrx/store';
import { AuthService } from '@services/auth.service';
import { AppState } from '@store/app.reducers';
import { Subscription } from 'rxjs';
import { unsetUsuarioActual } from '../store/actions/configuracion.action';

@Component({
  selector: 'app-navbar',
  templateUrl: './navbar.component.html',
  styleUrls: ['./navbar.component.scss'],
})
export class NavbarComponent implements OnInit, OnDestroy {
  private subscriptionStore: Subscription = new Subscription();
  nombreUsuario: string = '';

  constructor(
    private store: Store<AppState>,
    private router: Router,
    private authService: AuthService
  ) {
    // Se recibe el usuario actual almacenado en redux
    this.subscriptionStore = this.store
      .select('configuracion')
      .subscribe((data) => {
        this.nombreUsuario = data.usuarioActual?.usuario.nombre || '';
      });
  }

  ngOnInit(): void {}

  // Nos desuscribimos para evitar perdidas de memoria
  ngOnDestroy() {
    this.subscriptionStore.unsubscribe();
  }

  // Funcion que se llama para cerrar sesion
  salir() {

```

```

    this.authService.logout(); //Se llama a un service que le comunicara al back que se a
    cerrado sesion
    this.router.navigateByUrl('inicio'); //Se redirecciona al usuario a la landingpage
    this.store.dispatch(new unsetUsuarioActual()); // Se elimina usuario de redux
  }
}

```

### Service auth.component.ts logout

```

// Método que se comunica con el back para informar que el usuario cerró sesión
// Y que debe inhabilitar su token
logout() {
  this.http
    .post<any>(`${environment.url}/auth/logout`, null, {
      headers: this.headers,
    })
    .pipe(take(1));
}

```

HU N°8 Cambiar Email

Planilla de documentación

<b>Nombre de la HU</b>	Cambiar Email
<b>Módulo seleccionado</b>	Seguridad
<b>Descripción breve de la HU</b>	Como usuario de la aplicación quiero cambiar el mail.
<b>Actor</b>	Usuario
<b>Prioridad</b>	A
<b>Programador FrontEnd</b>	Juan Segundo Riera
<b>Programador BackEnd</b>	Miguel Del Corso
<b>Tester</b>	Jose Aranciba
<b>Aprobado por</b>	Elvio Aruta
<b>Denominación del programa</b>	<b>backend</b> "auth.controller.js" "auth.routes.js" <b>frontend</b> "perfil.component.ts" "perfilService.component.ts" "perfil.component.html"



<b>Fecha inicio Programación</b>	09/08/2021
<b>Fecha finalización de programación</b>	13/08/2021
<b>Base de datos</b>	MongoDB Redis
<b>Colecciones involucradas</b>	Usuario
<b>Historias de usuarios relacionadas</b>	HU 9 - Cambiar contraseña HU 10 - Cambiar contraseña desde email
<b>Casos de prueba relacionados con resultados</b>	
<p>Cambiar el mail de un usuario con el campo correo con un solo mail vacío=&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo correo con un mail ya asociado a un usuario=&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo correo con Un mail sin caracteres previos al arroba =&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo correo con el carácter: " =&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo correo con Un mail con más de un punto luego de el arroba =&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo correo con Un mail terminando en punto =&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo correo con Un mail sin puntos luego de el arroba =&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo correo con Un mail con caracteres especiales=&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo correo con Un mail con más de un @=&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo correo con Un mail con el carácter: -=&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo correo con Un mail con el carácter: _=&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo correo con Un mail con menos de 2 caracteres luego del punto final=&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo correo con Un mail con menos de 1 caracter luego del arroba y previo al punto=&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo correo con Un mail con más de 11 caracteres=&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo contraseña con una contraseña vacío=&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo contraseña con una contraseña incorrecta=&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo contraseña con una contraseña correcta=&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo contraseña con una contraseña anterior=&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo contraseña con un json mal formulado =&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo contraseña con los caracteres: \" =&gt; PASS</p> <p>Cambiar el mail de un usuario con el campo contraseña con Dos caracteres: \ =&gt; PASS</p>	

Documentación del BackEnd

Cambiar el email del usuario

- `${URL}/api/auth/cambiarEmail`
- Method: POST

### Body

```
{
  "nuevoCorreo": "vofagi2535@kembung.com",
  "pass": "123456"
}
```

### Response

```
{
  "message": "Se ha generado un link para cambiar su email, por favor haga click en link que le enviamos en su nuevo email"
}
```

### auth.routes.js

```
router.post(
  "/cambiarEmail",
  [
    validarJWT,
    check("nuevoCorreo", "El correo no es válido").isEmail(),
    check("pass", "La contraseña debe tener más de 6 caracteres").isLength({
      min: 6,
    }),
    validarCampos,
  ],
  asyncHandler(solicitarCambioEmail)
);
```

### auth.controller.js

Enviar email al nuevo correo del usuario para verificar que el correo le pertenezca

```
const solicitarCambioEmail = async
```

```
({usuarioAutenticado:{password,correo,nombre},body:{nuevoCorreo,pass}},res)=>{  
  
  // Buscamos que el nuevo correo no este usado por otro usuario  
  const usuario = await existUsuarioByEmailService(nuevoCorreo)  
  if(usuario){  
    throw new AlreadyExists("El correo ingresado ya existe por otra cuenta, por favor  
intente nuevamente")  
  }  
  
  // Verificamos la contraseña del usuario  
  if(!Usuario.comparePassword(pass,password)){  
    throw new BadCredentials("La contraseñas no coinciden, por favor vuelva a  
intentarlo")  
  }  
  
  // Generamos un token para cambiar el email  
  const {uid} = await saveCambiarEmailService(correo,nuevoCorreo);  
  
  // Enviamos un email al nuevo correo para hacer efectivo el cambio  
  await cambiarEmailService(nombre,nuevoCorreo,uid)  
  
  res.json({  
    message:  
      "Se ha generado un link para cambiar su email, por favor haga click en link que  
le enviamos en su nuevo email",  
  });  
}
```

Realizar el cambio de email en el usuario

```
const confirmarCambioEmail = async ({params:{token}},res) =>{  
  // Buscamos el token por el id que guardamos en redis  
  const {correo,nuevoCorreo} = await findTokenByIdService(token);  
  
  if (!correo || !nuevoCorreo) {  
    res.render("respuesta", {  
      layout: false,  
      message: "El link ha expirado, por favor vuelva a generar uno  
nuevo",  
    });  
  }  
  
  // Buscamos al usuario a través del correo que estaba guardado en el  
token de redis  
  const usuario = await findUsuarioActivoByEmailService(correo);  
  
  // Hacemos el cambio efectivo de mail  
  usuario.correo = nuevoCorreo
```

```
// Guardamos al usuario
await saveUsuarioService(usuario);

// Eliminamos el token de redis
await deleteTokenService(token);
res.render("respuesta", {
  layout: false,
  message: "El email se ha actualizado con éxito",
});
}
```

### Pantallas

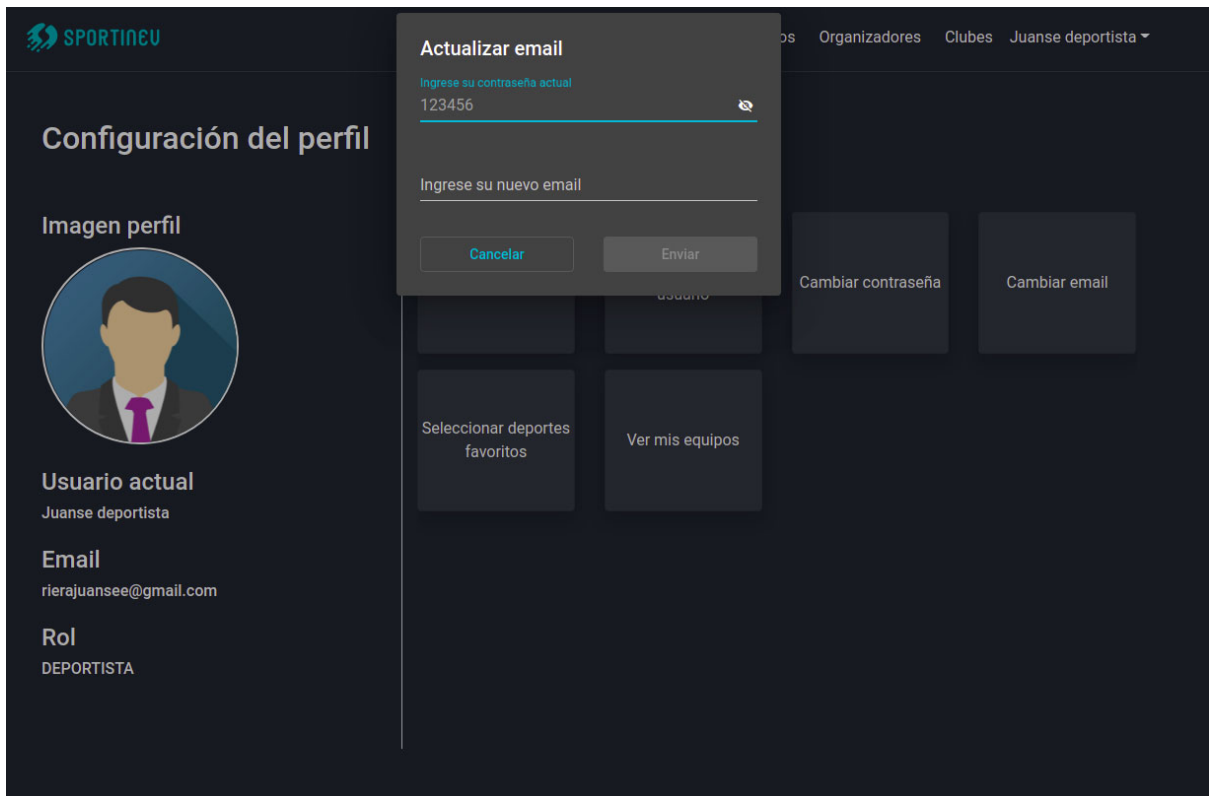


Figura 262: Pantalla de actualización de email.

### Documentación del FrontEnd

## Componente perfil.component.html

```

<!-- Información de pagina actual -->
<div class="row">
  <div class="col-12 align-center">
    <h2>Configuración del perfil</h2>
  </div>
</div>

<!-- Información del usuario logueado -->
<div class="row mt-5">
  <div class="col-sm-12 col-md-4 mb-5">

    <div class="mb-4">
      <h4>Imagen perfil</h4>
      <label class="hoverable" for="fileInputOriginal">
        <img [src]="datosPerfil?.img ? datosPerfil?.img :
'https://www.w3schools.com/howto/img_avatar.png'">
      </label>
    </div>

    <div class="my-4">
      <h4>Usuario actual</h4>
      <h6>{{datosPerfil?.nombre}}</h6>
    </div>

    <div class="my-4">
      <h4>Email</h4>
      <h6>{{datosPerfil?.correo}}</h6>
    </div>

    <div class="my-4">
      <h4>Rol</h4>
      <h6>{{rolPerfil}}</h6>
    </div>
  </div>

  <!-- Acciones que puede realizar el usuario
  Se setean desde ts y desde html genera un botón para cada una
  -->
  <div class="col-sm-12 col-md-8 border-left">

    <div *ngIf="acciones" id="scrollableContent">
      <div class="row">
        <ng-container *ngFor="let accion of acciones;">
          <div class="col-sm-12 col-md-5 col-lg-4 col-xl-3">
            <div id="card"
              class="card bg-dark shadow-sm mb-3 justify-content-center
              align-items-center text-center shadow"
              (click)="accion.operacion()">
              {{accion.nombre}}
            </div>
          </div>
        </ng-container>
      </div>
    </div>
  </div>

```

```
        </ng-container>
      </div>
    </div>
  </div>
</div>
```

### Componente perfil.component.ts

```
import { AppState } from '@store/app.reducers';
import { unsetUsuarioActual } from '../store/actions/configuracion.action';
import { DialogsService } from '../services/dialogs.service';
import { Subscription } from 'rxjs';
import { Router } from '@angular/router';
import { AuthService } from '@services/auth.service';
import { PerfilService } from '@services/perfil.service';

@Component({
  selector: 'app-perfil',
  templateUrl: './perfil.component.html',
  styleUrls: ['./perfil.component.scss'],
})
export class PerfilComponent implements OnInit, OnDestroy {
  private subscriptionStore: Subscription = new Subscription();
  acciones: Array<any> | null = null;
  datosPerfil: any = null;
  rolPerfil: any = null;

  constructor(
    private store: Store<AppState>,
    private dialogsService: DialogsService,
    private router: Router,
    private authService: AuthService,
    private perfilService: PerfilService
  ) {
    // Obtengo el usuario actual almacenado en redux y seteo las acciones que puede realizar
    this.subscriptionStore = this.store
      .select('configuracion')
      .subscribe((configuracionState) => {
        if (configuracionState.usuarioActual) {
          this.datosPerfil = configuracionState.usuarioActual.usuario;
          this.rolPerfil = configuracionState.usuarioActual.rol;
          this.cargarAccionesXRol(configuracionState.usuarioActual.rol);
          if (!this.acciones || this.acciones.length == 0) {
            // En caso de que no hayan acciones que pueda realizar, lo cual no debería
            // pasar
            // Se los redirecciona al home
            this.errorRedirect();
          }
        }
      });
  }
}
```

```
ngOnInit(): void {}

// Nos desuscribimos del redux para evitar pérdidas de memoria
ngOnDestroy() {
  this.subscriptionStore.unsubscribe();
}

// Funcion para cargar las acciones que pueden realizar los usuarios dependiendo de su rol
cargarAccionesXRol(rol: string) {
  //Se cargan acciones que puede realizar todo usuario
  let accionesACargar = this.getAccionesGenericas();
  // Se cargan las acciones por rol
  switch (rol) {
    case 'ADMINISTRADOR':
      accionesACargar = accionesACargar.concat(
        this.getAccionesAdministrador()
      );
      break;
    case 'DEPORTISTA':
      accionesACargar = accionesACargar.concat(this.getAccionesDeportista());
      break;
    case 'ARBITRO':
      accionesACargar = accionesACargar.concat(this.getAccionesArbitro());
      break;
    case 'PATROCINADOR':
      accionesACargar = accionesACargar.concat(
        this.getAccionesPatrocinador()
      );
      break;
    case 'ORGANIZADOR':
      accionesACargar = accionesACargar.concat(
        this.getAccionesAdministrador()
      );
      break;
    case 'CLUB':
      accionesACargar = accionesACargar.concat(this.getAccionesClub());
      break;

    default:
      //Cerrar sesion
      this.store.dispatch(new unsetUsuarioActual());
      break;
  }
  this.acciones = accionesACargar; //Se setean las acciones
}

// Acciones genericas de todos los usuarios
getAccionesGenericas(): Array<any> {
  return [
    // Modulo perfil
    {
      nombre: 'Actualizar datos',

```

```

    operacion: () => {
      this.router.navigate(['perfil/editar_datos']);
    },
  },
  // Modulo perfil
  {
    nombre: 'Crear nuevo tipo de usuario',
    operacion: () => {
      //TODO
    },
  },
  // Cambiar contraseña es del modulo de seguridad, cuando se presione en el html
  // Se llamara a La función que seteamos aca, La misma se explica más adelante
  {
    nombre: 'Cambiar contraseña',
    operacion: () => {
      this.cambiarContraseña();
    },
  },
  // Cambiar email es del modulo de seguridad, cuando se presione en el html
  // Se llamara a La función que seteamos aca, La misma se explica más adelante
  {
    nombre: 'Cambiar email',
    operacion: () => {
      this.cambiarEmail();
    },
  },
];
}

// Acciones del administrador TODO modulo de perfil
getAccionesAdministrador(): Array<any> {
  return [];
}

// Acciones del deportista TODO modulo perfil
getAccionesDeportista(): Array<any> {
  return [
    {
      nombre: 'Seleccionar deportes favoritos',
      operacion: () => {
        //TODO
      },
    },
  ],
  {
    nombre: 'Ver mis equipos',
    operacion: () => {
      this.router.navigate(['deportista/ver_equipos']);
    },
  },
];
}

```



```
// Acciones del organizador TODO módulo perfil
getAccionesOrganizador(): Array<any> {
  return [
    {
      nombre: 'Comprar cuenta premium',
      operacion: () => {
        //TODO
      },
    },
  ];
}

// Acciones de Los clubes TODO módulo perfil
getAccionesClub(): Array<any> {
  return [];
}

// Acciones de Los arbitro TODO módulo de perfil
getAccionesArbitro(): Array<any> {
  return [];
}

// Accioens de Los patrocinadores TODO módulo de perfil
getAccionesPatrocinador(): Array<any> {
  return [];
}

// En caso de un error se Le informa al usuario y se Lo redirige a home
errorRedirect() {
  this.dialogsService.showError(
    'Se produjo un error, vuelva a intentarlo mas tarde'
  );
  this.router.navigateByUrl('inicio');
}

// Funcion cambiar contraseña usa un service que llamara a un modal
// que no permitira obtener la información necesaria para cambiar contraseña
cambiarContraseña() {
  this.dialogsService
    .cambiarContrasenna()
    .afterClosed()
    .subscribe((res) => {
      if (res) {
        this.dialogsService.cargando('Cambiando contraseña');
        // Se envia al service La contraseña actual y La nueva para que se La envíe al
back
        this.perfilService
          .cambiarContraseña(res.contrasennaActual, res.nuevaContrasenna)
          .subscribe(
            (respuesta) => {
              // En caso de obtener una respuesta correcta se vuelve a setear el
usuario en el redux
              // y se Le informa al usuario que todo funciono correctamente
            }
          )
        }
      }
    )
  }
}
```

```

        this.authService.actualizarUsuarioActual();
        this.dialogsService.cerrar();
        this.dialogsService.toastSuccess(
            'Contraseña cambiada correctamente'
        );
    },
    (error) => {
        this.dialogsService.cerrar();
        this.dialogsService.toastError(
            'Se produjo un error, vuelva a intentarlo más tarde'
        );
    }
    });
}
}

// Funcion cambiar email usa un service que llamara a un modal
// que no permitira obtener la información necesaria para cambiar email
cambiarEmail() {
    this.dialogsService
        .cambiarEmail()
        .afterClosed()
        .subscribe((res) => {
            if (res) {
                this.dialogsService.cargando('Cambiando email');
                // Se envia al service la contraseña actual y el nuevo email que desea tener el
                // usuario
                this.perfilService
                    .cambiarEmail(res.contrasennaActual, res.emailNuevo)
                    .subscribe(
                        (respuesta) => {
                            // En caso de que todo salga bien se lo saca al usuario del sistema
                            // y se le pide que se vuelva a logear
                            this.dialogsService.cerrar();
                            this.dialogsService.success(
                                '',
                                'Se ha enviado un link a su nuevo email, para que los cambios tengan
                                efecto debe meterse al mismo y luego vuelva a logearse'
                            );
                            this.store.dispatch(new unsetUsuarioActual());
                        },
                        (error) => {
                            this.dialogsService.cerrar();
                            this.dialogsService.toastError(
                                'Se produjo un error, vuelva a intentarlo más tarde'
                            );
                        }
                    );
            }
        });
}
}
}

```

### Service perfilService.component.ts

```
// API para cambiar email, hay que enviar contraseña actual y nuevo email deseado
// junto con el token del usuario en el header
cambiarEmail(pass: string, nuevoCorreo: string) {
  return this.http
    .post<any>(
      `${environment.url}/auth/cambiarEmail`,
      {
        pass,
        nuevoCorreo,
      },
      {
        headers: this.headers,
      }
    )
    .pipe(take(1));
}
```

### HU N° 9 Cambiar Contraseña

#### Planilla de documentación

<b>Nombre de la HU</b>	Cambiar Contraseña
<b>Módulo seleccionado</b>	Seguridad
<b>Descripción breve de la HU</b>	Como usuario de la aplicación quiero cambiar la contraseña desde mi perfil para poder aumentar la seguridad y privacidad del usuario.
<b>Actor</b>	Usuario
<b>Prioridad</b>	A
<b>Programador BackEnd</b>	Miguel Del Corso
<b>Programador FrontEnd</b>	Juan Riera
<b>Tester</b>	Jose Aranciba
<b>Aprobado por</b>	Elvio Aruta
<b>Denominación del programa</b>	<b>backend</b> "auth.controller.js" "auth.routes.js" <b>frontend</b>

	"perfil.component.ts" "perfilService.component.ts" "perfil.component.html"
<b>Fecha inicio Programación</b>	09/08/2021
<b>Fecha finalización de programación</b>	13/08/2021
<b>Base de datos</b>	MongoDB Redis
<b>Colecciones involucradas</b>	Usuario
<b>Historias de usuarios relacionadas</b>	HU 8 - Cambiar email HU 10 - Cambiar contraseña desde email
<b>Casos de prueba relacionados con resultados</b>	
<p>Cambiar la contraseña de un usuario con el campo contraseña nueva con una contraseña vacío=&gt; PASS</p> <p>Cambiar la contraseña de un usuario con el campo contraseña nueva con una contraseña incorrecta=&gt; PASS</p> <p>Cambiar la contraseña de un usuario con el campo contraseña nueva con una contraseña correcta=&gt; PASS</p> <p>Cambiar la contraseña de un usuario con el campo contraseña nueva con una contraseña anterior=&gt; PASS</p> <p>Cambiar la contraseña de un usuario con el campo contraseña nueva con un json mal formulado =&gt; PASS</p> <p>Cambiar la contraseña de un usuario con el campo contraseña nueva con los caracteres: \ =&gt; PASS</p> <p>Cambiar la contraseña de un usuario con el campo contraseña nueva con Dos caracteres: \ =&gt; PASS</p> <p>Cambiar la contraseña de un usuario con el campo contraseña actual con una contraseña vacío=&gt; PASS</p> <p>Cambiar la contraseña de un usuario con el campo contraseña actual con una contraseña incorrecta=&gt; PASS</p> <p>Cambiar la contraseña de un usuario con el campo contraseña actual con una contraseña correcta=&gt; PASS</p> <p>Cambiar la contraseña de un usuario con el campo contraseña actual con una contraseña anterior=&gt; PASS</p> <p>Cambiar la contraseña de un usuario con el campo contraseña actual con un json mal formulado =&gt; PASS</p> <p>Cambiar la contraseña de un usuario con el campo contraseña actual con los caracteres: \ =&gt; PASS</p> <p>Cambiar la contraseña de un usuario con el campo contraseña actual con Dos caracteres: \ =&gt; PASS</p> <p>Cambiar la contraseña de un usuario con el campo contraseña actual y contraseña nueva iguales =&gt; PASS</p>	

Documentación del BackEnd

Cambiar contraseña actual

Un usuario autenticado en el sistema desea cambiar su email.

**URL:** `${URL}/api/auth/cambiarContraseniasActual`

**Method:** POST

### Body

```
{
  "contraseniaActual": "123456",
  "nuevaContrasenia": "987654"
}
```

### Response

```
{
  "message": "La contraseña se ha actualizado con éxito"
}
```

### auth.routes.js

```
router.post("/cambiarContraseniaActual", [
  validarJWT,
  check("contraseniaActual", "La contraseña debe tener más de 6 caracteres").isLength({min:6,}),
  check("nuevaContrasenia", "La nueva contraseña debe tener más de 6 caracteres").isLength({min:6,}),
  validarCampos
], asyncHandler(cambiarContraseniaActual))
```

### auth.controller.js

```
const cambiarContraseniaActual = async
({usuarioAutenticado:{correo,password},body:{contraseniaActual,nuevaContrasenia}},res)=>
{
  if(contraseniaActual === nuevaContrasenia){
    throw new GeneralError("La contraseña actual ingresada y la nueva son la misma")
  }

  // verificamos que la contraseña actual coincida
  if(!Usuario.comparePassword(contraseniaActual,password)){
    throw new BadCredentials("La contraseñas no coinciden, por favor vuelva a
```

```
intentarlo")
}

// Buscamos al usuario a través del correo
const usuario = await findUsuarioActivoByEmailService(correo)

// Encriptamos la contraseña
usuario.password = await Usuario.encryptPassword(nuevaContraseña);

// Guardamos al usuario
await saveUsuarioService(usuario);

res.json({
  message: "La contraseña se ha actualizado con éxito",
});
}
```

## Pantallas

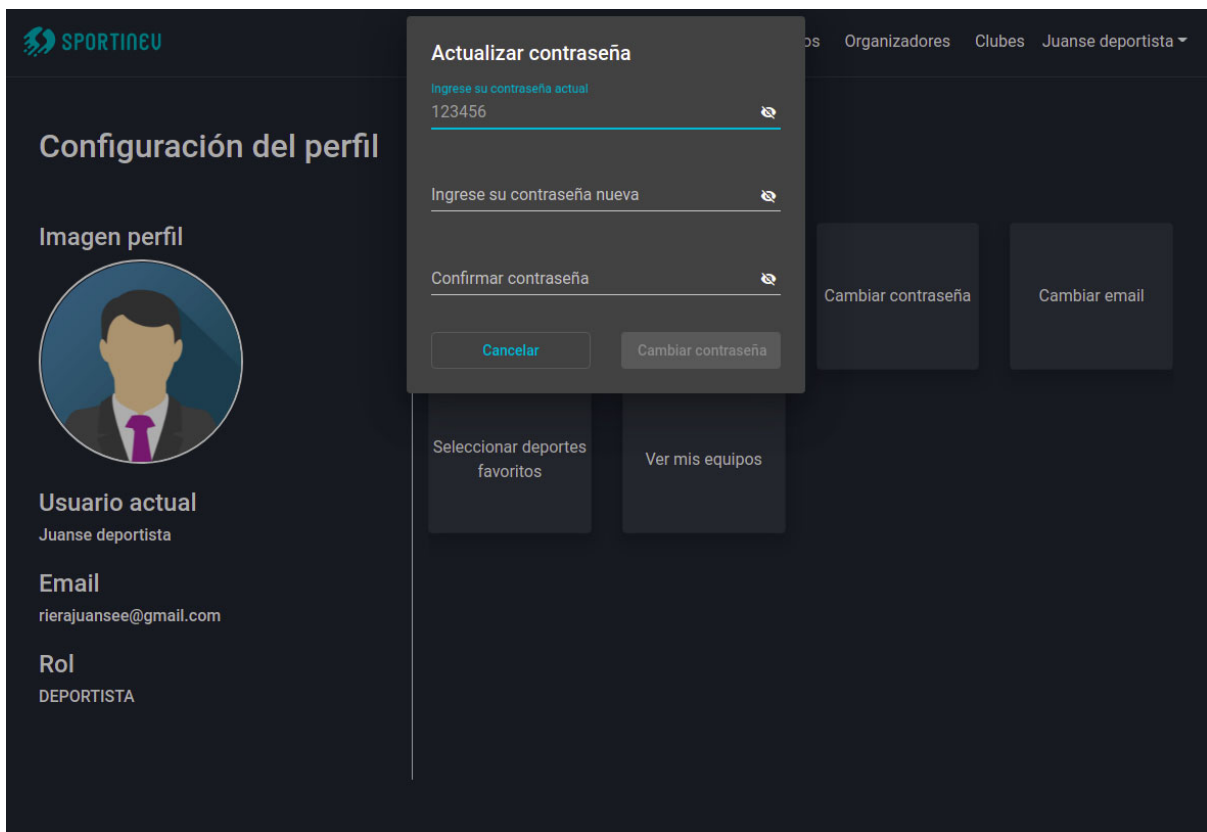


Figura 263: Pantalla de recuperación de contraseña.

## Documentación del FrontEnd

## Componente perfil.component.html

```

<!-- Información de pagina actual -->
<div class="row">
  <div class="col-12 align-center">
    <h2>Configuración del perfil</h2>
  </div>
</div>

<!-- Información del usuario logueado -->
<div class="row mt-5">
  <div class="col-sm-12 col-md-4 mb-5">

    <div class="mb-4">
      <h4>Imagen perfil</h4>
      <label class="hoverable" for="fileInputOriginal">
        <img [src]="datosPerfil?.img ? datosPerfil?.img :
'https://www.w3schools.com/howto/img_avatar.png'">
      </label>
    </div>

    <div class="my-4">
      <h4>Usuario actual</h4>
      <h6>{{datosPerfil?.nombre}}</h6>
    </div>

    <div class="my-4">
      <h4>Email</h4>
      <h6>{{datosPerfil?.correo}}</h6>
    </div>

    <div class="my-4">
      <h4>Rol</h4>
      <h6>{{rolPerfil}}</h6>
    </div>
  </div>

  <!-- Acciones que puede realizar el usuario
  Se setean desde ts y desde html genera un botón para cada una
  -->
  <div class="col-sm-12 col-md-8 border-left">

    <div *ngIf="acciones" id="scrollableContent">
      <div class="row">
        <ng-container *ngFor="let accion of acciones;">
          <div class="col-sm-12 col-md-5 col-lg-4 col-xl-3">
            <div id="card"
              class="card bg-dark shadow-sm mb-3 justify-content-center
              align-items-center text-center shadow"
              (click)="accion.operacion()">
              {{accion.nombre}}
            </div>
          </div>
        </ng-container>
      </div>
    </div>
  </div>

```

```
        </ng-container>
      </div>
    </div>
  </div>
</div>
```

### Componente perfil.component.ts

```
import { Component, OnInit, OnDestroy } from '@angular/core';
import { Store } from '@ngrx/store';
import { AppState } from '@store/app.reducers';
import { unsetUsuarioActual } from '../store/actions/configuracion.action';
import { DialogsService } from '../services/dialogs.service';
import { Subscription } from 'rxjs';
import { Router } from '@angular/router';
import { AuthService } from '@services/auth.service';
import { PerfilService } from '@services/perfil.service';

@Component({
  selector: 'app-perfil',
  templateUrl: './perfil.component.html',
  styleUrls: ['./perfil.component.scss'],
})
export class PerfilComponent implements OnInit, OnDestroy {
  private subscriptionStore: Subscription = new Subscription();
  acciones: Array<any> | null = null;
  datosPerfil: any = null;
  rolPerfil: any = null;

  constructor(
    private store: Store<AppState>,
    private dialogsService: DialogsService,
    private router: Router,
    private authService: AuthService,
    private perfilService: PerfilService
  ) {
    // Obtengo el usuario actual almacenado en redux y seteo las acciones que puede realizar
    this.subscriptionStore = this.store
      .select('configuracion')
      .subscribe((configuracionState) => {
        if (configuracionState.usuarioActual) {
          this.datosPerfil = configuracionState.usuarioActual.usuario;
          this.rolPerfil = configuracionState.usuarioActual.rol;
          this.cargarAccionesXRol(configuracionState.usuarioActual.rol);
          if (!this.acciones || this.acciones.length == 0) {
            // En caso de que no hayan acciones que pueda realizar, lo cual no debería pasar
            // Se los redirecciona al home
            this.errorRedirect();
          }
        }
      });
  }
}
```



```
}  
  
ngOnInit(): void {}  
  
// Nos desuscribimos del redux para evitar pérdidas de memoria  
ngOnDestroy() {  
  this.subscriptionStore.unsubscribe();  
}  
  
// Funcion para cargar las acciones que pueden realizar los usuarios dependiendo de su rol  
cargarAccionesXRol(rol: string) {  
  //Se cargan acciones que puede realizar todo usuario  
  let accionesACargar = this.getAccionesGenericas();  
  // Se cargan las acciones por rol  
  switch (rol) {  
    case 'ADMINISTRADOR':  
      accionesACargar = accionesACargar.concat(  
        this.getAccionesAdministrador()  
      );  
      break;  
    case 'DEPORTISTA':  
      accionesACargar = accionesACargar.concat(this.getAccionesDeportista());  
      break;  
    case 'ARBITRO':  
      accionesACargar = accionesACargar.concat(this.getAccionesArbitro());  
      break;  
    case 'PATROCINADOR':  
      accionesACargar = accionesACargar.concat(  
        this.getAccionesPatrocinador()  
      );  
      break;  
    case 'ORGANIZADOR':  
      accionesACargar = accionesACargar.concat(  
        this.getAccionesAdministrador()  
      );  
      break;  
    case 'CLUB':  
      accionesACargar = accionesACargar.concat(this.getAccionesClub());  
      break;  
  
    default:  
      //Cerrar sesion  
      this.store.dispatch(new unsetUsuarioActual());  
      break;  
  }  
  this.acciones = accionesACargar; //Se setean las acciones  
}  
  
// Acciones genericas de todos los usuarios  
getAccionesGenericas(): Array<any> {  
  return [  
    // Modulo perfil
```

```
{
  nombre: 'Actualizar datos',
  operacion: () => {
    this.router.navigate(['perfil/editar_datos']);
  },
},
// Modulo perfil
{
  nombre: 'Crear nuevo tipo de usuario',
  operacion: () => {
    //TODO
  },
},
// Cambiar contraseña es del modulo de seguridad, cuando se presione en el html
// Se llamara a La función que seteamos aca, La misma se explica más adelante
{
  nombre: 'Cambiar contraseña',
  operacion: () => {
    this.cambiarContraseña();
  },
},
// Cambiar email es del modulo de seguridad, cuando se presione en el html
// Se llamara a La función que seteamos aca, La misma se explica más adelante
{
  nombre: 'Cambiar email',
  operacion: () => {
    this.cambiarEmail();
  },
},
];
}

// Acciones del administrador TODO modulo de perfil
getAccionesAdministrador(): Array<any> {
  return [];
}

// Acciones del deportista TODO modulo perfil
getAccionesDeportista(): Array<any> {
  return [
    {
      nombre: 'Seleccionar deportes favoritos',
      operacion: () => {
        //TODO
      },
    },
    {
      nombre: 'Ver mis equipos',
      operacion: () => {
        this.router.navigate(['deportista/ver_equipos']);
      },
    },
  ];
};
```

```
}

// Acciones del organizador TODO módulo perfil
getAccionesOrganizador(): Array<any> {
  return [
    {
      nombre: 'Comprar cuenta premium',
      operacion: () => {
        //TODO
      },
    },
  ];
}

// Acciones de Los clubes TODO módulo perfil
getAccionesClub(): Array<any> {
  return [];
}

// Acciones de Los arbitro TODO módulo de perfil
getAccionesArbitro(): Array<any> {
  return [];
}

// Accioens de Los patrocinadores TODO módulo de perfil
getAccionesPatrocinador(): Array<any> {
  return [];
}

// En caso de un error se Le informa al usuario y se Lo redirige a home
errorRedirect() {
  this.dialogsService.showError(
    'Se produjo un error, vuelva a intentarlo mas tarde'
  );
  this.router.navigateByUrl('inicio');
}

// Funcion cambiar contraseña usa un service que llamara a un modal
// que no permitira obtener la información necesaria para cambiar contraseña
cambiarContraseña() {
  this.dialogsService
    .cambiarContrasenna()
    .afterClosed()
    .subscribe((res) => {
      if (res) {
        this.dialogsService.cargando('Cambiando contraseña');
        // Se envia al service la contraseña actual y la nueva para que se la envíe al
back
        this.perfilService
          .cambiarContraseña(res.contrasennaActual, res.nuevaContrasenna)
          .subscribe(
            (respuesta) => {
              // En caso de obtener una respuesta correcta se vuelve a setear el
```

```

usuario en el redux
    // y se le informa al usuario que todo funciono correctamente
    this.authService.actualizarUsuarioActual();
    this.dialogsService.cerrar();
    this.dialogsService.toastSuccess(
        'Contraseña cambiada correctamente'
    );
  },
  (error) => {
    this.dialogsService.cerrar();
    this.dialogsService.toastError(
        'Se produjo un error, vuelva a intentarlo más tarde'
    );
  }
});
}
});
}

// Funcion cambiar email usa un service que llamara a un modal
// que no permitira obtener la información necesaria para cambiar email
cambiarEmail() {
  this.dialogsService
    .cambiarEmail()
    .afterClosed()
    .subscribe((res) => {
      if (res) {
        this.dialogsService.cargando('Cambiano email');
        // Se envia al service la contraseña actual y el nuevo email que desea tener el
        usuario
        this.perfilService
          .cambiarEmail(res.contrasennaActual, res.emailNuevo)
          .subscribe(
            (respuesta) => {
              // En caso de que todo salga bien se lo saca al usuario del sistema
              // y se le pide que se vuelva a logear
              this.dialogsService.cerrar();
              this.dialogsService.success(
                '',
                'Se ha enviado un link a su nuevo email, para que los cambios tengan
                efecto debe meterse al mismo y luego vuelva a logearse'
              );
              this.store.dispatch(new unsetUsuarioActual());
            },
            (error) => {
              this.dialogsService.cerrar();
              this.dialogsService.toastError(
                'Se produjo un error, vuelva a intentarlo más tarde'
              );
            }
          );
      }
    });
}
});
}
});
}

```

```
}  
}
```

### Service perfilService.component.ts

```
// Se comunica con La API para cambiar La contraseña actual  
// se debe enviar contraseña actual y nueva contraseña  
// junto con el token del usuario en el header  
cambiarContraseña(contraseniaActual: string, nuevaContrasenia: string) {  
  return this.http  
    .post<any>(  
      `${environment.url}/auth/cambiarContraseniaActual`,  
      {  
        contraseniaActual,  
        nuevaContrasenia,  
      },  
      {  
        headers: this.headers,  
      }  
    )  
    .pipe(take(1));  
}
```

HU N°10 Cambiar Contraseña desde email

Planilla de documentación

<b>Nombre de la HU</b>	Cambiar Contraseña desde email
<b>Módulo seleccionado</b>	Seguridad
<b>Descripción breve de la HU</b>	Como usuario de la aplicación quiero recuperar la contraseña desde la página de logueo para poder loguearme nuevamente al sistema o aumentar la seguridad de mi cuenta.
<b>Actor</b>	Usuario
<b>Prioridad</b>	A
<b>Programador BackEnd</b>	Miguel Del Corso
<b>Programador FrontEnd</b>	Juan Riera
<b>Tester</b>	Elvio Aruta

<b>Aprobado por</b>	Ignacio Starkloff
<b>Denominación del programa</b>	<b>backend</b> "auth.controller.js" "auth.routes.js" <b>frontend</b> "perfil.component.ts" "perfilService.component.ts" "perfil.component.html"
<b>Fecha inicio Programación</b>	09/08/2021
<b>Fecha finalización de programación</b>	13/08/2021
<b>Base de datos</b>	MongoDB Redis
<b>Colecciones involucradas</b>	Usuario
<b>Historias de usuarios relacionadas</b>	HU 8 - Cambiar email HU 9 - Cambiar Contraseña
<b>Casos de prueba relacionados con resultados</b>	
<p>Cambiar contraseña desde el mail de un usuario con el campo correo con un solo mail vacío=&gt; PASS</p> <p>Cambiar contraseña desde el mail de un usuario con el campo correo con un mail ya asociado a un usuario=&gt; PASS</p> <p>Cambiar contraseña desde el mail de un usuario con el campo correo con Un mail sin caracteres previos al arroba =&gt; PASS</p> <p>Cambiar contraseña desde el mail de un usuario con el campo correo con el carácter: " =&gt; PASS</p> <p>Cambiar contraseña desde el mail de un usuario con el campo correo con Un mail con más de un punto luego de el arroba =&gt; PASS</p> <p>Cambiar contraseña desde el mail de un usuario con el campo correo con Un mail terminando en punto =&gt; PASS</p> <p>Cambiar contraseña desde el mail de un usuario con el campo correo con Un mail sin puntos luego de el arroba =&gt; PASS</p> <p>Cambiar contraseña desde el mail de un usuario con el campo correo con Un mail con caracteres especiales=&gt; PASS</p> <p>Cambiar contraseña desde el mail de un usuario con el campo correo con Un mail con más de un @=&gt; PASS</p> <p>Cambiar contraseña desde el mail de un usuario con el campo correo con Un mail con el carácter: -=&gt; PASS</p> <p>Cambiar contraseña desde el mail de un usuario con el campo correo con Un mail con el carácter: _=&gt; PASS</p> <p>Cambiar contraseña desde el mail de un usuario con el campo correo con Un mail con menos de 2 caracteres luego del punto final=&gt; PASS</p> <p>Cambiar contraseña desde el mail de un usuario con el campo correo con Un mail con menos de 1 caracter luego del arroba y previo al punto=&gt; PASS</p> <p>Cambiar contraseña desde el mail de un usuario con el campo correo con Un mail con más de 11 caracteres=&gt; PASS</p>	

Documentación del BackEnd

### Solicitar Cambio de contraseña

El usuario solicita un cambio de contraseña desde el login.

- **URL:** `${URL}/api/auth/cambiarContrasenna`
- **Method:** POST

#### Body

```
{  
  "correo": "starkloffnacho@gmail.com"  
}
```

#### Response

```
{  
  "message": "Se ha generado un link para cambiar su contraseña, por favor verifique su  
email y haga click en el link"  
}
```

### Mostrar Template Cambiar Contraseña

Al hacer click en el link que le llegó al mail del usuario que solicitó la operación, se muestra un template que permita realizar el cambio de contraseña.

- **URL:** `${URL}/api/auth/cambiarContrasenna/{token}`
- **Method:** GET

### Confirmar Cambio Contraseña

Efectuar el cambio de contraseña.

- **URL:** `${URL}/api/auth/cambiarContrasenna`
- **Method:** PUT

#### Body

```
{  
  "resetToken": "c3RhcmVsb2ZmbmFjaG9AZ21haWwY29t",  
  "password": "123456"  
}
```

## Response

```
{  
  "message": "La contraseña se ha actualizado con éxito"  
}
```

## auth.routes.js

```
router.get("/cambiarContrasenna/:token", asyncHandler(mostrarCambioContrasennaForm));  
router.post("/cambiarContrasenna", asyncHandler(solicitarCambiarContrasenna));  
router.put("/cambiarContrasenna", asyncHandler(confirmCambioContrasenna));
```

## auth.controller.js

```
const mostrarCambioContrasennaForm = async (req = request, res = response) => {  
  try {  
    const idToken = req.params.token;  
    console.log(idToken);  
    const token = await findTokenByIdService(idToken);  
  
    if (!token || !token.correo) {  
      res.render("respuesta", {  
        layout: false,  
        message: "El link ha expirado, por favor vuelva a generar uno nuevo",  
      });  
    }  
  
    res.render("cambioContrasenna", {  
      resetToken: idToken,  
      layout: false,  
    });  
  } catch (error) {  
    console.log(error);  
    res.render("respuesta", {  
      layout: false,  
      message:  
        "Error, hubo un problema que no permite cambiar su contraseña, por favor  
        comuníquese con el administrador",  
    });  
  }  
}
```



```
}  
};
```

```
const confirmarCambioContrasenna = async (req = request, res = response) => {  
  const { password, resetToken } = req.body;  
  // Buscamos el token por su id el cual fue pasado por el body  
  const token = await findTokenByIdService(resetToken);  
  // Si el token no existe o no corresponde a un token destinado a cambiar la contraseña  
  if (!token || !token.correo) {  
    throw new NotFound("El link no es válido para cambiar la contraseña");  
  }  
  // Buscamos al usuario a través del correo que estaba guardado en el token en redis  
  const usuario = await findUsuarioActivoByEmailService(token.correo);  
  // Encriptamos la contraseña  
  usuario.password = await Usuario.encryptPassword(password);  
  // Guardamos al usuario  
  await saveUsuarioService(usuario);  
  // Eliminamos el token  
  await deleteTokenService(resetToken);  
  res.json({  
    message: "La contraseña se ha actualizado con éxito",  
  });  
};
```

```
const confirmarCambioContrasenna = async (req = request, res = response) => {  
  const { password, resetToken } = req.body;  
  // Buscamos el token por su id el cual fue pasado por el body  
  const token = await findTokenByIdService(resetToken);  
  // Si el token no existe o no corresponde a un token destinado a cambiar la contraseña  
  if (!token || !token.correo) {  
    throw new NotFound("El link no es válido para cambiar la contraseña");  
  }  
  // Buscamos al usuario a través del correo que estaba guardado en el token en redis  
  const usuario = await findUsuarioActivoByEmailService(token.correo);  
  // Encriptamos la contraseña  
  usuario.password = await Usuario.encryptPassword(password);  
  // Guardamos al usuario  
  await saveUsuarioService(usuario);  
  // Eliminamos el token  
  await deleteTokenService(resetToken);  
  res.json({  
    message: "La contraseña se ha actualizado con éxito",  
  });  
};
```

## Pantalla

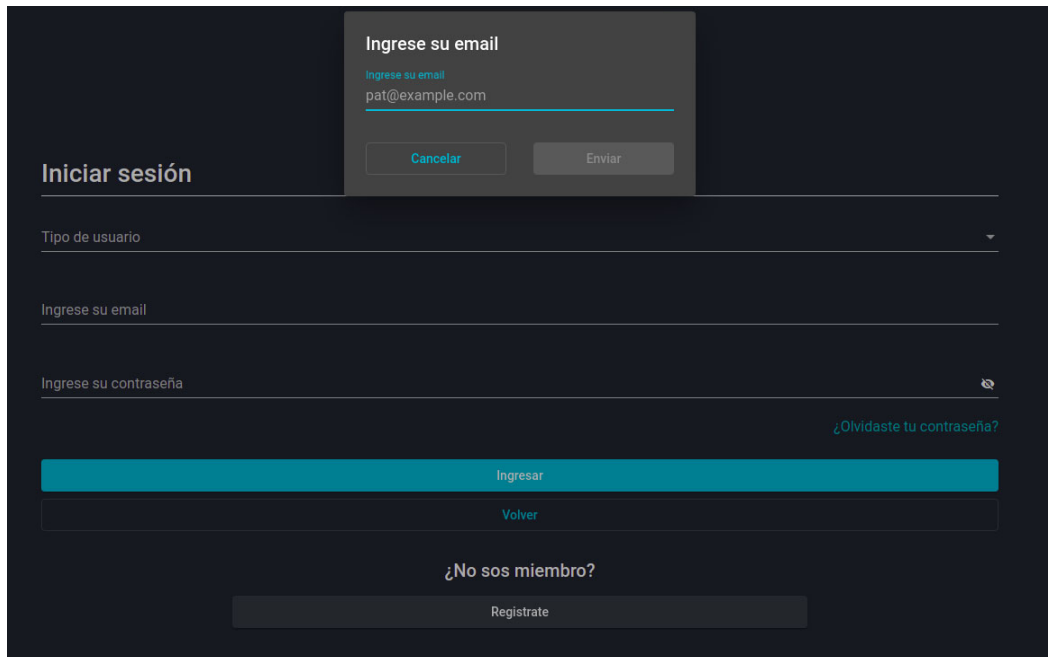


Figura 264: Pantalla de recuperación de contraseña.

## Documentación del FrontEnd

### Componente login.component.html

```

<div class="container mt-5">
  <div class="row">
    <!-- Logo de la aplicación -->
    <div class="col-12 text-center">
      
    </div>
  </div>
  <!-- Componente card nos da una estructura responsive solo hay que pasarle el form -->
  <app-card-form [componentePasado]=loginForm titulo="Iniciar sesión"></app-card-form>
</div>

<!-- Form que se le enviara a card -->
<ng-template #loginForm>
  <form [formGroup]="form" (ngSubmit)="enviarInformacion()">

    <!-- Input para obtener tipo de rol del usuario -->
    <div class="my-3">
      <mat-form-field appearance="legacy" style="width: 100%;">
        <mat-label>Tipo de usuario</mat-label>
        <mat-select formControlName="rol">
          <ng-container *ngFor="let rol of roles">
            <mat-option [value]=rol>
  
```

```

        {{rol}}
      </mat-option>
    </ng-container>
  </mat-select>
</mat-form-field>
</div>

<!-- Input para obtener email del usuario -->
<div class="my-3">
  <mat-form-field appearance="legacy" style="width: 100%;">
    <mat-label>Ingrese su email</mat-label>
    <input matInput placeholder="pat@example.com" formControlName="email"
autocomplete="off">
    <mat-error *ngIf="form.controls['email'].errors">El
      email ingresado no es válido</mat-error>
    </mat-form-field>
  </div>

  <!-- Input para obtener contraseña del usuario -->
  <div>
    <mat-form-field appearance="legacy" style="width: 100%;">
      <mat-label>Ingrese su contraseña</mat-label>
      <input matInput [type]="hide ? 'password' : 'text'" placeholder="123456"
formControlName="password"
      autocomplete="off">
      <button mat-icon-button matSuffix (click)="hide = !hide" [attr.aria-label]="'Hide
password'"
        [attr.aria-pressed]="hide" type="button">
        <mat-icon>{{hide ? 'visibility_off' : 'visibility'}}</mat-icon>
      </button>
      <mat-error *ngIf="form.controls['password'].errors">La contraseña debe tener al
menos 6
        caracteres</mat-error>
    </mat-form-field>
  </div>

  <!-- Link para abrir modal para recuperar contraseña -->
  <p class="text-right mb-4">
    <a (click)="olvidoContrasenna()" class="cursor">¿Olvidaste tu contraseña?</a>
  </p>

  <!-- Botones para logearse o para volver a la landing page -->
  <div class="row mt-3">
    <div class="col-12 text-center mb-2">
      <button mat-raised-button color="primary" style="width: 100%;">Ingresar</button>
    </div>feedback
    <div class="col-12 text-center mb-2">
      <button mat-stroked-button color="primary" style="width: 100%;"(
click)="volver()">Volver</button>
    </div>
  </div>

  <!-- Redireccionar a usuario a pagina de signup -->

```

```
<h5 class="text-center mt-4">¿No sos miembro?</h5>
<div class="row mt-3">
  <div class="col-12 text-center">
    <button mat-raised-button class="btn-dark" style="width: 60%;"
(click)="irASignup()">Registrar</button>
  </div>
</div>
</form>
</ng-template>
```

### Componente login.component.ts

```
//Se envia mail para que actualizar contraseña
recuperarContraseña(email: string) {
  return this.http
    .post<any>(`${environment.url}/auth/cambiarContrasenna`, {
      correo: email,
    })
    .pipe(take(1));
}
```

### Service auth.component.ts

```
//Se envia mail para que actualizar contraseña
recuperarContraseña(email: string) {
  return this.http
    .post<any>(`${environment.url}/auth/cambiarContrasenna`, {
      correo: email,
    })
    .pipe(take(1));
}
```

HU N° 11 Contactarse con el administrador del sistema

Planilla de documentación

<b>Nombre de la HU</b>	Contactarse con el administrador del sistema
<b>Módulo seleccionado</b>	Seguridad
<b>Descripción breve de la HU</b>	Como usuario quiero poder contactarme con el administrador del sistema para poder informarle sobre errores que se han producido al usar la aplicación o pedirle nuevas funcionalidades.

<b>Actor</b>	Usuario
<b>Prioridad</b>	A
<b>Programador BackEnd</b>	Ignacio Starkloff
<b>Programador FrontEnd</b>	Juan Riera
<b>Tester</b>	Elvio Aruta
<b>Aprobado por</b>	Jose Aranciba
<b>Denominación del programa</b>	<b>frontend</b> "auth.component.ts" "contactanos.component.ts" "contactanos.component.html"
<b>Fecha inicio Programación</b>	09/08/2021
<b>Fecha finalización de programación</b>	13/08/2021
<b>Base de datos</b>	MongoDB
<b>Colecciones involucradas</b>	-
<b>Historias de usuarios relacionadas</b>	HU 6 - LogIn HU 7 - Log Out HU 9 - Cambiar Contraseña
<b>Casos de prueba relacionados con resultados</b>	
<p>contactarse con el administrador con el campo nombre con un array vacío=&gt; PASS  contactarse con el administrador con el campo nombre con caracteres especiales=&gt; PASS  contactarse con el administrador con el campo nombre con Un entero =&gt; PASS  contactarse con el administrador con el campo nombre con Un número con coma =&gt; PASS  contactarse con el administrador con el campo nombre con Un número con punto =&gt; PASS  contactarse con el administrador con el campo nombre con Un espacio en blanco=&gt; PASS  contactarse con el administrador con el campo nombre con un json mal formulado =&gt; PASS  contactarse con el administrador con el campo nombre con los caracteres: \" =&gt; PASS  contactarse con el administrador con el campo nombre con Dos caracteres: \ =&gt; PASS  contactarse con el administrador con el campo nombre con menos de 11 caracteres : \ =&gt; PASS  contactarse con el administrador con el campo apellido con un array vacío=&gt; PASS  contactarse con el administrador con el campo apellido con caracteres especiales=&gt; PASS  contactarse con el administrador con el campo apellido con Un entero =&gt; PASS  contactarse con el administrador con el campo apellido con Un número con coma =&gt;</p>	

PASS

contactarse con el administrador con el campo apellido con Un número con punto => PASS

contactarse con el administrador con el campo apellido con Un espacio en blanco=> PASS

contactarse con el administrador con el campo apellido con un json mal formulado => PASS

contactarse con el administrador con el campo apellido con los caracteres: \" => PASS

contactarse con el administrador con el campo apellido con Dos caracteres: \ => PASS

contactarse con el administrador con el campo teléfono con el campo vacío=> PASS

contactarse con el administrador con el campo teléfono con un número correcto=> PASS

contactarse con el administrador con el campo teléfono con el número cero=> PASS

contactarse con el administrador con el campo teléfono con un número con coma=> PASS

contactarse con el administrador con el campo teléfono con un número con punto=> PASS

contactarse con el administrador con el campo teléfono con un número negativo=> PASS

contactarse con el administrador con el campo teléfono con una expresión matemática=> PASS

contactarse con el administrador con el campo teléfono con una cadena de caracteres alfabéticos=> PASS

contactarse con el administrador con el campo teléfono con un carácter alfabético=> PASS

contactarse con el administrador con el campo teléfono con un carácter alfabético entre enteros=> PASS

contactarse con el administrador con el campo teléfono con un carácter especial=> PASS

contactarse con el administrador con el campo teléfono con más de 60 cifras=> PASS

contactarse con el administrador con el campo teléfono con null=> PASS

contactarse con el administrador con el campo teléfono con Nan=> PASS

contactarse con el administrador con el campo teléfono con NaN=> PASS

contactarse con el administrador con el campo teléfono con Na=> PASS

contactarse con el administrador con el campo teléfono con false=> PASS

contactarse con el administrador con el campo teléfono con true=> PASS

contactarse con el administrador con el campo email con un solo mail vacío=> PASS

contactarse con el administrador con el campo email con un mail ya asociado a un usuario=> PASS

contactarse con el administrador con el campo email con Un mail sin caracteres previos al arroba => PASS

contactarse con el administrador con el campo email con el carácter: " => PASS

contactarse con el administrador con el campo email con Un mail con más de un punto luego de el arroba => PASS

contactarse con el administrador con el campo email con Un mail terminando en punto => PASS

contactarse con el administrador con el campo email con Un mail sin puntos luego de el arroba => PASS

contactarse con el administrador con el campo email con Un mail con caracteres especiales=> PASS

contactarse con el administrador con el campo email con Un mail con más de un @=> PASS

contactarse con el administrador con el campo email con Un mail con el carácter: -=> PASS

contactarse con el administrador con el campo email con Un mail con el carácter: \_=> PASS  
contactarse con el administrador con el campo email con Un mail con menos de 2 caracteres luego del punto final=> PASS  
contactarse con el administrador con el campo email con Un mail con menos de 1 caracter luego del arroba y previo al punto=> PASS  
contactarse con el administrador con el campo email con Un mail con más de 100 caracteres=> PASS  
contactarse con el administrador con el campo mensaje con un array vacío=> PASS  
contactarse con el administrador con el campo mensaje con caracteres especiales=> PASS  
contactarse con el administrador con el campo mensaje con Un entero => PASS  
contactarse con el administrador con el campo mensaje con Un número con coma => PASS  
contactarse con el administrador con el campo mensaje con Un número con punto => PASS  
contactarse con el administrador con el campo mensaje con Un espacio en blanco=> PASS  
contactarse con el administrador con el campo mensaje con un json mal formulado => PASS  
contactarse con el administrador con el campo mensaje con los caracteres: \" => PASS  
contactarse con el administrador con el campo mensaje con Dos caracteres: \ => PASS

## Pantalla



The screenshot shows a registration form for Sportinev. At the top, there is a logo for Sportinev, which consists of a stylized blue and red figure resembling a person or a hand, followed by the text 'SPORTINEV' in blue. Below the logo, the text reads: 'Por favor complete el siguiente cuestionario para que nuestros analistas lo revisen'. The form contains five input fields, each with a label above it: 'Ingrese su nombre', 'Ingrese su apellido', 'Ingrese su teléfono', 'Ingrese su email', and 'Mensaje para administrador'. At the bottom of the form, there are two buttons: a prominent blue button labeled 'Registrarse' and a smaller, light blue button labeled 'Volver'.

Figura 265: Pantalla de contacto.

Documentación FrontEnd

### Componente contactanos.component.html

```
import { Component, OnInit } from '@angular/core';
import {
  FormBuilder,
  FormControl,
  FormGroup,
  Validators,
} from '@angular/forms';
import { Router } from '@angular/router';
import { blankSpaceValidator } from '@helpers/blankspaceValidator';
import { DialogsService } from '@services/dialogs.service';
import { AuthService } from '../services/auth.service';

@Component({
  selector: 'app-contactanos',
  templateUrl: './contactanos.component.html',
  styleUrls: ['./contactanos.component.scss'],
})
export class ContactanosComponent implements OnInit {
  hide: boolean = true;
  // Mensaje que se muestra en la página por defecto, puede ser sobrescrito
  // si se envía un nuevo título desde el componente que lo este llamando
  mensaje: string =
    'Por favor complete el siguiente cuestionario para que nuestros analistas lo
    revisen';
  error: any = 'No hay información sobre el error';

  // Información básica para enviar al administrador
  form: FormGroup = new FormGroup({
    nombre: new FormControl('', [Validators.required, blankSpaceValidator]),
    apellido: new FormControl('', [Validators.required, blankSpaceValidator]),
    contexto: new FormControl('', [
      Validators.required,
      blankSpaceValidator,
      Validators.maxLength(999),
    ]),
    telefono: new FormControl('', [
      Validators.required,
      blankSpaceValidator,
      Validators.pattern('[0-9]{7,10}'),
    ]),
    email: new FormControl('', [Validators.required, Validators.email]),
  });

  constructor(
    private router: Router,
```



```
public FormBuilder: FormBuilder,  
private dialogsService: DialogsService,  
private authService: AuthService  
) {  
  // En caso de errores graves en la aplicación se le ofrecerá al usuario comunicarse  
  // con el administrador en caso que decida enviar información se envía data sobre que  
  // endpoint provocó el error, además se puede usar un mensaje más personalizado para  
  el  
  // tipo de error en caso de enviar algo  
  if (this.router.getCurrentNavigation()?.extras?.state) {  
    let parameters = this.router.getCurrentNavigation()?.extras.state;  
  
    if (parameters?.mensajeError) {  
      this.mensaje = parameters.mensajeError;  
    }  
  
    if (parameters?.error) {  
      this.error = parameters.error;  
    }  
  
    if (parameters?.showAlert.showAlert) {  
      this.dialogsService.error('', parameters.showAlert.mensajeAlert);  
    }  
  }  
}  
  
ngOnInit(): void { }  
  
enviarInformacion() {  
  this.dialogsService.cargando("Enviando información, por favor espere");  
  if (this.form.invalid) {  
    return;  
  }  
  this.authService.contactarseConAdministrador(this.form.getRawValue()).subscribe(  
    (respuesta)=>{  
      // En caso de una respuesta correcta se le informa al usuario que todo salio  
      correctamente  
      // y se lo redirecciona a la landing page  
      this.dialogsService.cerrar();  
      this.dialogsService.toastSuccess("Se a enviado información al adminitrador  
correctamente");  
      this.router.navigateByUrl("inicio");  
    },  
    (error)=>{  
      // En caso de error se le pide al usuario que lo vuelva a intentar más tarde  
      // Y se lo redirecciona a la landingpage  
      this.dialogsService.cerrar();  
      this.dialogsService.toastError(  
        'No se pudo enviar información, vuelva a intentarlo más tarde'  
      );  
      this.router.navigateByUrl('inicio');  
    }  
  )  
}
```

```
}  
  
// Acción llamada desde el botón del html para volver a La Landing page  
volver() {  
  this.router.navigate(['inicio']);  
}  
}
```

### Componente contactanos.component.ts

```
import { Component, OnInit } from '@angular/core';  
import {  
  FormBuilder,  
  FormControl,  
  FormGroup,  
  Validators,  
} from '@angular/forms';  
import { Router } from '@angular/router';  
import { blankSpaceValidator } from '@helpers/blankspaceValidator';  
import { DialogsService } from '@services/dialogs.service';  
import { AuthService } from '../services/auth.service';  
  
@Component({  
  selector: 'app-contactanos',  
  templateUrl: './contactanos.component.html',  
  styleUrls: ['./contactanos.component.scss'],  
})  
export class ContactanosComponent implements OnInit {  
  hide: boolean = true;  
  // Mensaje que se muestra en la página por defecto, puede ser sobrescrito  
  // si se envía un nuevo título desde el componente que lo este llamando  
  mensaje: string =  
    'Por favor complete el siguiente cuestionario para que nuestros analistas lo  
  revisen';  
  error: any = 'No hay información sobre el error';  
  
  // Información básica para enviar al administrador  
  form: FormGroup = new FormGroup({  
    nombre: new FormControl('', [Validators.required, blankSpaceValidator]),  
    apellido: new FormControl('', [Validators.required, blankSpaceValidator]),  
    contexto: new FormControl('', [  
      Validators.required,  
      blankSpaceValidator,  
      Validators.maxLength(999),  
    ]),  
    telefono: new FormControl('', [  
      Validators.required,  
      blankSpaceValidator,  
      Validators.pattern('[0-9]{7,10}'),  
    ]),  
    email: new FormControl('', [Validators.required, Validators.email]),  
  });  
}
```

```
});

constructor(
  private router: Router,
  public formBuilder: FormBuilder,
  private dialogsService: DialogsService,
  private authService: AuthService
) {
  // En caso de errores graves en la aplicación se le ofrecerá al usuario comunicarse
  // con el administrador en caso que decida enviar información se envía data sobre que
  // endpoint provocó el error, además se puede usar un mensaje más personalizado para
  // el
  // tipo de error en caso de enviar algo
  if (this.router.getCurrentNavigation()?.extras?.state) {
    let parameters = this.router.getCurrentNavigation()?.extras.state;

    if (parameters?.mensajeError) {
      this.mensaje = parameters.mensajeError;
    }

    if (parameters?.error) {
      this.error = parameters.error;
    }

    if (parameters?.showAlert.showAlert) {
      this.dialogsService.error('', parameters.showAlert.mensajeAlert);
    }
  }
}

ngOnInit(): void { }

enviarInformacion() {
  this.dialogsService.cargando("Enviando información, por favor espere");
  if (this.form.invalid) {
    return;
  }
  this.authService.contactarseConAdministrador(this.form.getRawValue()).subscribe(
    (respuesta)=>{
      // En caso de una respuesta correcta se le informa al usuario que todo salio
      // correctamente
      // y se lo redirecciona a la landing page
      this.dialogsService.cerrar();
      this.dialogsService.toastSuccess("Se a enviado información al adminitrador
      correctamente");
      this.router.navigateByUrl("inicio");
    },
    (error)=>{
      // En caso de error se le pide al usuario que lo vuelva a intentar más tarde
      // Y se lo redirecciona a la landingpage
      this.dialogsService.cerrar();
      this.dialogsService.toastError(
        'No se pudo enviar información, vuelva a intentarlo más tarde'
      );
    }
  );
}
```

```
);  
  this.router.navigateByUrl('inicio');  
}  
)  
  
}  
  
// Acción llamada desde el botón del html para volver a La Landing page  
volver() {  
  this.router.navigate(['inicio']);  
}  
}
```

### Service auth.component.ts

```
contactarseConAdministrador(datos:any){  
  //Se envia información a administrador  
  // No es necesario estar logueado  
  return this.http  
    .post<any>(environment.url + '/controlar/contactarse',datos, {  
      headers: this.headers,  
    })  
    .pipe(take(1));  
}
```

## Estructura de la documentación de backend

Los “end points” que comunican el front end con el back end y la base de datos para transmitir la información necesaria a mostrar al usuario, están documentados en la plataforma GitLab respetando la siguiente estructura:

### Readme Principal

Se detallan en este menú del proyecto, los requerimientos del sistema, el procedimiento para instalar el mismo una vez descargado, para correrlo en diferentes ambientes, y principalmente las APIs de cada uno de sus módulos describiendo las rutas pertenecientes a cada una de ellas como se ve a continuación:

### APIS Sportinev

---

Sportinev cuenta con las siguientes apis

- [Auth routes](#) /api/auth/
- [Partidos routes](#) /api/partidos/
- [Usuarios routes](#) /api/usuarios
- [Eventos routes](#) /api/eventos
- [Brackets routes](#) /api/brackets
- [Controlar acción routes](#) /api/controlar
- [Canchas routes](#) /api/canchas
- [Departamentos routes](#) /api/departamentos
- [Provincias routes](#) /api/provincias
- [Localidades routes](#) /api/localidades
- [Países routes](#) /api/paises
- [Deportes routes](#) /api/deportes
- [Categorías routes](#) /api/categorias
- [Tipo Organizador routes](#) /api/tipoOrganizador
- [Global routes](#) /api/global
- [Permisos routes](#) /api/permisos
- [Perfil routes](#) /api/perfil
- [Roles routes](#) /api/permisos
- [Clubes routes](#) /api/clubes
- [Tipo eventos routes](#) /api/tipoEventos
- [Redes sociales](#) /api/redes
- [Inscripciones](#) /api/inscripcion
- [Backups](#) /api/backup
- [Feedback](#) /api/feedback
- [Reportes](#) /api/reportes

Figura 266: Documentación de apis en readme

## Endpoint de cada Módulo

En cada una de las páginas de estos enlaces detallados previamente, podemos observar las funcionalidades de cada módulo detalladas por URL y tipo de request necesario para ejecutar la funcionalidad. Esta documentación de las funcionalidades del módulo sigue la siguiente estructura:

### Auth routes

*Título del módulo padre*

#### SignIn

*{Título del módulo}*

Identifica en el sistema al usuario con un correo, contraseña y rol específico.

*{Descripción del módulo}.*

URL  $\${URL}/api/auth/signIn$

URL  $\${URL}/api/{Modulo Padre}/{módulo}$

Method POST

Method {método a utilizar para la llamada}

```

{
  "correo": "vofagi2535@gmail.com",
  "password": "789456",
  "rol": "DEPORTISTA"
}
  
```

*{Body del request}*

```

{
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9",
  "usuario": {
    "fechaInicioVig": "2021-08-19T14:06:36.647Z",
    "_id": "611e6570f3139924224264c4",
    "nombre": "Miguel",
    "correo": "vofagi2535@kembung.com",
    "estados": [
      {
        "fechaInicioVig": "2021-08-19T14:06:36.646Z",
        "_id": "611e6570f3139924224264c6",
      }
    ]
  }
}
  
```

```

        "estado": "6115fc6c8fd6a830f52b2c8f"
      }
    ],
    "roles": [
      {
        "fechaInicioVig": "2021-08-19T14:06:36.646Z",
        "_id": "611e6570f3139924224264c5",
        "rol": "6113601985887dfb2d950257"
      }
    ],
    "deportista": {
      "fechaInicioVig": "2021-08-19T14:06:36.673Z",
      "_id": "611e6570f3139924224264c9",
      "apellido": "Miguel Del Corso",
      "usuario": "611e6570f3139924224264c4",
      "deportes": [],
      "__v": 0
    }
  },
  "rol": "DEPORTISTA"
}

```

*Resultado Esperado*

**Consecutivamente**, se escriben las demás operaciones del módulo siguiendo la misma sintaxis.

El ejemplo completo puede ser visualizado en repositorio del sistema.

## Planificación de capacitación

### Introducción

El plan de capacitación tiene como objetivo principal el hecho de proveer los conocimientos necesarios para un correcto uso de Sportinev para los usuarios finales.

### Capacitación para los usuarios

#### Objetivos de la capacitación

Los objetivos de la capacitación son:

- Fomentar el interés y la participación de las personas en la aplicación.

- Capacitar al usuario en el uso del sistema y establecer las líneas básicas que debe seguir para la correcta utilización del sistema.
- Agilizar la creación de los eventos.
- Detallar el alcance de cada uno de los reportes que provee el sistema.
- Colaborar con los usuarios de la aplicación en la utilización de la misma.

### Destinatarios

El Plan de Capacitación incluye a todos los potenciales usuarios del sistema, y tiene una duración de 5 días. El mismo comenzará durante la etapa de implementación después de que el sistema esté desplegado en el ambiente productivo.

### Temas de capacitación

Los temas más importantes a tratar durante la capacitación serán:

- Introducción a los roles del sistema (árbitro, deportista, organizador, patrocinador, club).
- Capacitación del administrador del sistema para dar de alta nuevas entidades
- Parámetro.
- Capacitación para llevar a cabo las funcionalidades de creación de eventos.
- Ventajas de la utilización del sistema.

### Recursos a emplear

Los recursos a emplear son:

### Materiales

- Manual de usuario.
- Breve tutorial inicial con slides para el usuario estándar.
- Apartado de preguntas frecuentes en el sistema.
- Computadora con internet, equipo de sonido, celulares Android.

### Humanos

- Participantes especializados del proyecto. Se decidieron 4 personas que estarán presentes en la capacitación: Líder del proyecto, tester, programador, analista de sistema.

### Estrategias o modalidades a utilizar

Las estrategias a emplear para poder efectuar la capacitación:

- **Reunión de capacitación para la explicación del sistema y sus objetivos:** Se organiza una reunión con quienes serán los usuarios de Sportinev para explicar el



funcionamiento del sistema de forma general, proyectando el sistema en funcionamiento y realizando las funciones básicas.

- **Taller interactivo:** Se realizan simulaciones prácticas del uso del sistema desde la perspectiva del rol de:
  - Desde la perspectiva del rol de administrador realizando las tareas de configuración del sistema.
  - Árbitro: para centrarse en la forma de registrar los resultados del partido y como realiza la confirmación del pago.
  - Deportista: para centrarse en la forma de inscripción a un evento.
  - Organizador: para centrarse en la creación de eventos.
  - Patrocinador: para centrarse en la forma de promoción de un evento.
  - Club: para poder aceptar / rechazar eventos.
- **Lecturas del manual de usuario:** Se realizan de forma paralela a la reunión de capacitación y a los talleres para poder aclarar las dificultades que surjan en el uso del sistema.

### Cronograma de capacitación

La duración total de la preparación de la capacitación en el uso del sistema será de 5 días, la cual se encuentra prevista en el Anexo N°1: Diagrama de Gantt.

Número	Tarea	Duración (Horas)
1	Elaboración de manual de usuario	100
3	Elaboración de slides para el manual de módulo de seguridad	4
4	Elaboración de slides para el manual de módulo de eventos	3
5	Elaboración de slides para el manual de módulo de reportes	2
6	Elaboración de slides para el manual de módulo de publicidad	1
7	Elaboración de slides para el manual de módulo de inscripción	2
8	Elaboración de slides para el manual de módulo de pagos	3
9	Elaboración de slides para el manual de módulo de Feedback	1
10	Elaboración de slides para el manual de módulo de parametrización	1
11	Elaboración de slides para el manual de módulo Perfil	3
12	Elaboración de apartado de preguntas frecuentes	4
13	Capacitación al administrador del sistema	5
<b>Total</b>		129

### Capacitación para el administrador del sistema

La capacitación para los administradores se llevará a través de reuniones virtuales, en las cuales se realizará:

	Tareas
Día N°	Administrador
1	Manejo de la sección Perfil, cerrar sesión, darse de baja.
2	Manejo de: tipos de organizadores, estado inscripción evento, estado evento, estado patrocinador del evento, tipos de deportes, tipos de eventos.
3	Alta, baja y modificación de deportes Dar de alta un rol con sus permisos. Asignar roles de administrador a otro usuario

## Planificación, ejecución y documentación de pruebas

En esta sección se presentan los resultados de las pruebas que se llevan a cabo para validar el correcto funcionamiento del sistema.

### Planeación de pruebas

#### Alcance

- Se planificarán las pruebas todos los requisitos funcionales de los módulos del sistema
- Se realizarán las pruebas de estrés y carga
- Se realizarán pruebas de rendimiento
- Se realizará la automatización de las pruebas que se consideren sumamente necesarias

#### Fuera de alcance

- No se realizarán las pruebas unitarias del todo código, ya que esto depende de cada desarrollador y la prioridad de la tarea
- No se realizarán pruebas de penetración

## Roles y responsabilidades

### Diseño de Test Cases

- Jose Aranciba
- Miguel Del Corso

### Ejecución de los Test Cases

- Jose Aranciba
- Miguel Del Corso
- Elvio Aruta
- Ignacio Starkloff
- Riera Juan Segundo

### Automatización de test cases

- Jose Aranciba
- Elvio Aruta

## Metodología de prueba

Se utilizará una metodología ágil para realizar los tests para aquellas funcionalidades en desarrollo.

La metodología de testing ágil se efectúa testeando cada entregable evitando que las próximas entregas se realicen con bugs anteriores.

Se emplea esta metodología, ya que los cambios son bienvenidos en cualquier etapa del sprint al ser flexible y adaptarse a los requerimientos del cliente, y porque el testing incremental minimiza los riesgos posibles.

## Creación de datos de prueba

Cada una de las entidades/clases/tablas de prueba ingresadas en la base de datos o no, deben poseer un identificador único relacionado a las pruebas. El valor de este ID es:

test-sp<n°> - <id>

Considerando:

<n°> = como número de sprint

<id> = id generado automáticamente

## Estructura general de los casos de prueba

Los casos de prueba respetarán la siguiente estructura de tabla pudiendo ser modificado si es necesario añadiendo o quitando celdas en caso de aplicar.

ID		
Nombre		
Historia de usuario		
descripción		
Precondiciones		
Datos de prueba		
Fecha Ejecución		
tipo		
Pasos	Usuario	Sistema
Evidencia:		
<b>ESTADO</b>	<b>OKAY</b>	

### Duración de las pruebas

- Pruebas Unitarias:
  - programación: 50 días
  - revisión: 5 días
- Prueba de seguridad
  - diseño: 28 días
  - ejecución: 7 días
- Prueba de integración
  - diseño: 21 días
  - ejecución: 7 días
- Pruebas de carga
  - diseño: 2 días
  - ejecución: 1 días
- Pruebas de estrés
  - diseño: 2 días
  - ejecución: 2 días

### Actividades

Los tipos de prueba a usar en la aplicación serán:

- Pruebas unitarias
- Prueba de seguridad

- Prueba de integración
- Pruebas de carga
- Pruebas de estrés

### Complejidad del test

- Cobertura de condiciones
- N° de Test cases manuales y automatizados ejecutados
- Cantidad de defectos encontrados
- Defectos abiertos a ser reparados y defectos a reparar en la próxima entrega

### Entregables

- Test Plan
- Test Cases
- Bug Reports

### Herramientas de testing

- Gestor de incidencias: Trello
- Herramientas de pruebas de integración sistema : Jmeter, Postman
- Herramienta de gestión de pruebas unitarias: Jasmine
- Herramienta de automatización de pruebas: Jmeter
- Registro de Test Cases: Google Docs
- Herramienta de gestión de casos de prueba : Google Docs

### Recursos del entorno de pruebas

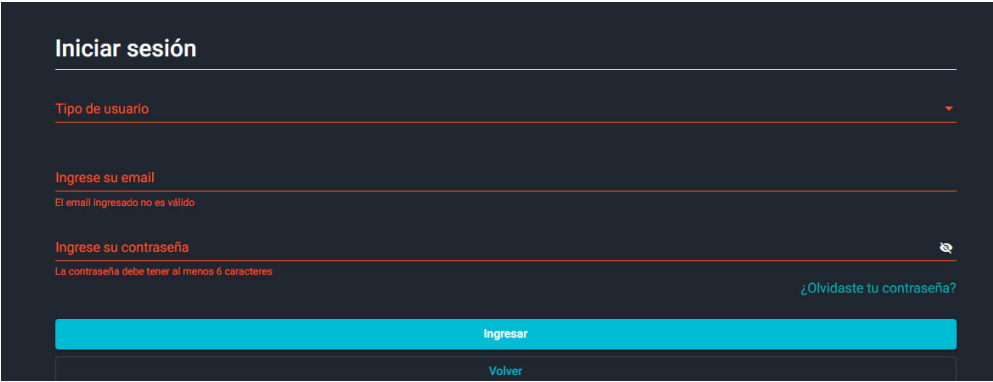
- Cualquier Sistema Operativo
- Google docs
- Jmeter
- Postman
- Mongo Atlas

### Pruebas Unitarias


Las pruebas unitarias son pruebas realizadas en porciones pequeñas de código, a continuación se detallan las pruebas unitarias para la historia de usuario : "Log in".

#### Test Cases Unitarias:

ID	TCU-01
Nombre	Ingresar sin datos

Historia de usuario	login	
descripción	El usuario intenta loguearse sin ingresar ningún dato	
Precondiciones	Usuario en página de ingreso	
Datos de prueba	-	
Fecha Ejecución	28/08/2021	
tipo	Negativa	
Pasos	Usuario	Sistema
	No ingresa ningún dato en los campos Clickea en el botón ingresar	
Resultado esperado	Muestra los errores de datos vacíos en los campos email, contraseña y tipo de usuario	
Evidencia:		
<b>ESTADO</b>	<b>OKAY</b>	

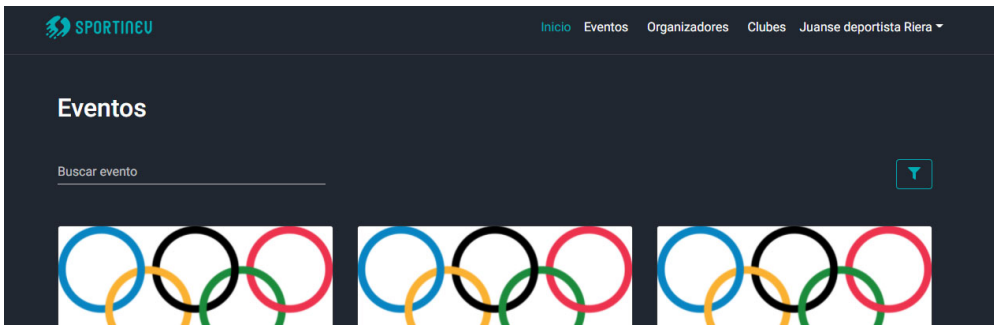
ID	TCU-02
Nombre	Ingresar con usuario inexistente
Historia de usuario	login
Descripción	El usuario intenta loguearse con un usuario no existente como árbitro.
Precondicio	Usuario en página de ingreso

nes		
Datos de prueba	mail: <a href="mailto:a@a.com">a@a.com</a> contraseña: 123456 Tipo de Usuario: Árbitro	
Fecha Ejecución	28/08/2021	
Tipo	Negativa	
Pasos	Usuario	Sistema
	<p>Clickea en el combobox tipo usuario</p> <p>Selecciona Árbitro</p> <p>Clickea en el campo mail</p> <p>Ingresa el mail :a@a.com</p> <p>Clickea el campo contraseña e ingresa la contraseña : 123456</p> <p>Clickea en el botón ingresar</p>	<p>Se muestran los tipo usuarios existentes y habilitados</p> <p>Se muestra el color azul en el campo indicando la selección correcta</p> <p>Se muestra un ejemplo del formato dentro del campo</p> <p>Se muestra el color azul en el campo indicando el formato correcto</p>
Resultado esperado	Muestra un cartel rojo con un mensaje de error respecto al usuario o contraseña erróneas.	
Evidencia:		

ESTADO	OKAY
--------	------

ID	TCU-03	
Nombre	Ingresar con usuario Deportista existente	
Historia de usuario	login	
Descripción	El usuario se loguea como juanse riera como deportista con éxito	
Precondiciones	Usuario en página de ingreso	
Datos de prueba	mail: rierajuansee@gmail.com contraseña: juanse Tipo de Usuario: Deportista	
Fecha Ejecución	28/08/2021	
tipo	Positiva	
Pasos	Usuario	Sistema
	<p>Clickea en el combobox tipo usuario</p> <p>Selecciona Deportista</p> <p>Clickea en el campo mail</p> <p>Ingresar el mail :rierajuansee@gmail.com</p> <p>Clickea el campo contraseña e ingresa la contraseña : juanse</p> <p>Clickea en el botón ingresar</p>	<p>Se muestran los tipo usuarios existentes y habilitados</p> <p>Se muestra el color azul en el campo indicando la selección correcta</p> <p>Se muestra un ejemplo del formato dentro del campo</p> <p>Se muestra el color azul en el campo indicando el formato correcto</p>
Resultado esperado	El sistema redirige al usuario y lo Ingresar a la página principal de eventos con el usuario logueado: "Juanse Deportista Riera"	
Evidencia:		



	
ESTADO	OKAY

Se llevó a cabo una implementación de pruebas unitarias automatizadas para agilizar el proceso de desarrollo. El código implementado es el siguiente:

```

import { ComponentFixture, TestBed } from '@angular/core/testing';

import { LoginComponent } from './login.component';

describe('LoginComponent', () => {
  let component: LoginComponent;
  let fixture: ComponentFixture<LoginComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      declarations: [ LoginComponent ]
    })
    .compileComponents();
  });

  beforeEach(() => {
    fixture = TestBed.createComponent(LoginComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });

  it(`should have as title 'Iniciar sesión'`, () => {
    const fixture = TestBed.createComponent(LoginComponent);
    const app = fixture.debugElement.componentInstance;

    expect(app.title).toEqual('Iniciar sesión');
  });

  it(`should verify email`, () => {
    const element: HTMLElement = fixture.nativeElement;
    const nameInput: HTMLInputElement =
  
```

```

element.querySelector('input[formControlName="email"]')!;
  const boton: HTMLElement = element.querySelector('button[test-id="button-login"]')!;
  const errormail: HTMLElement = element.querySelector('[test-id="errormail"]')!;
  nameInput.value = 'josearanciba09';
  nameInput.dispatchEvent(new Event('input'));
  boton.click();
  fixture.detectChanges();
  expect(errormail).toContain('email ingresado no es válido');

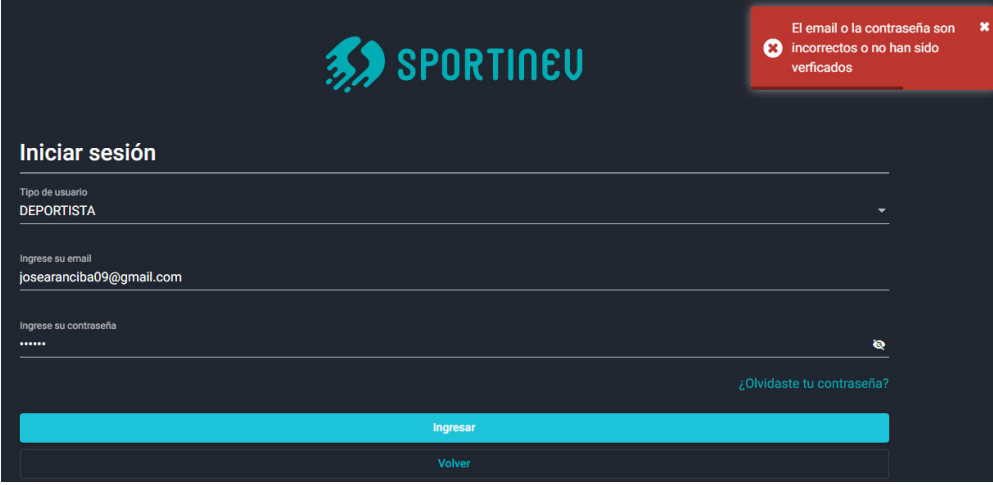
});
it('should throw pass error', () => {
  const element: HTMLElement = fixture.nativeElement;
  const boton: HTMLElement = element.querySelector('button[test-id="button-login"]')!;
  const errorpass: HTMLElement = element.querySelector('[test-id="errormail"]')!;
  boton.click();
  fixture.detectChanges();
  expect(errorpass).toContain('La contraseña debe tener al menos 6 caracteres');

});
});

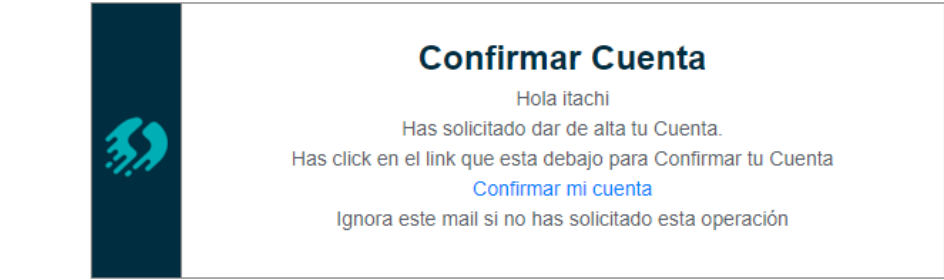
```

## Pruebas de Seguridad

ID	TCS-01	
Nombre	Loguearme como Deportista con usuario patrocinador	
Historia de usuario	Log in	
Descripción	El usuario intenta loguearse con un rol no habilitado para ese perfil	
Precondiciones	Usuario en página de registro, usuario creado con rol patrocinador	
Datos de prueba	mail: josearanciba09@gmail.com contraseña: 123456 Tipo de Usuario: Deportista	
Fecha Ejecución	28/08/2021	
tipo	Negativa	
Pasos	Usuario	Sistema
	Clickea en el combobox tipo usuario	Se muestran los tipo usuarios existentes y habilitados  Se muestra el color azul en el

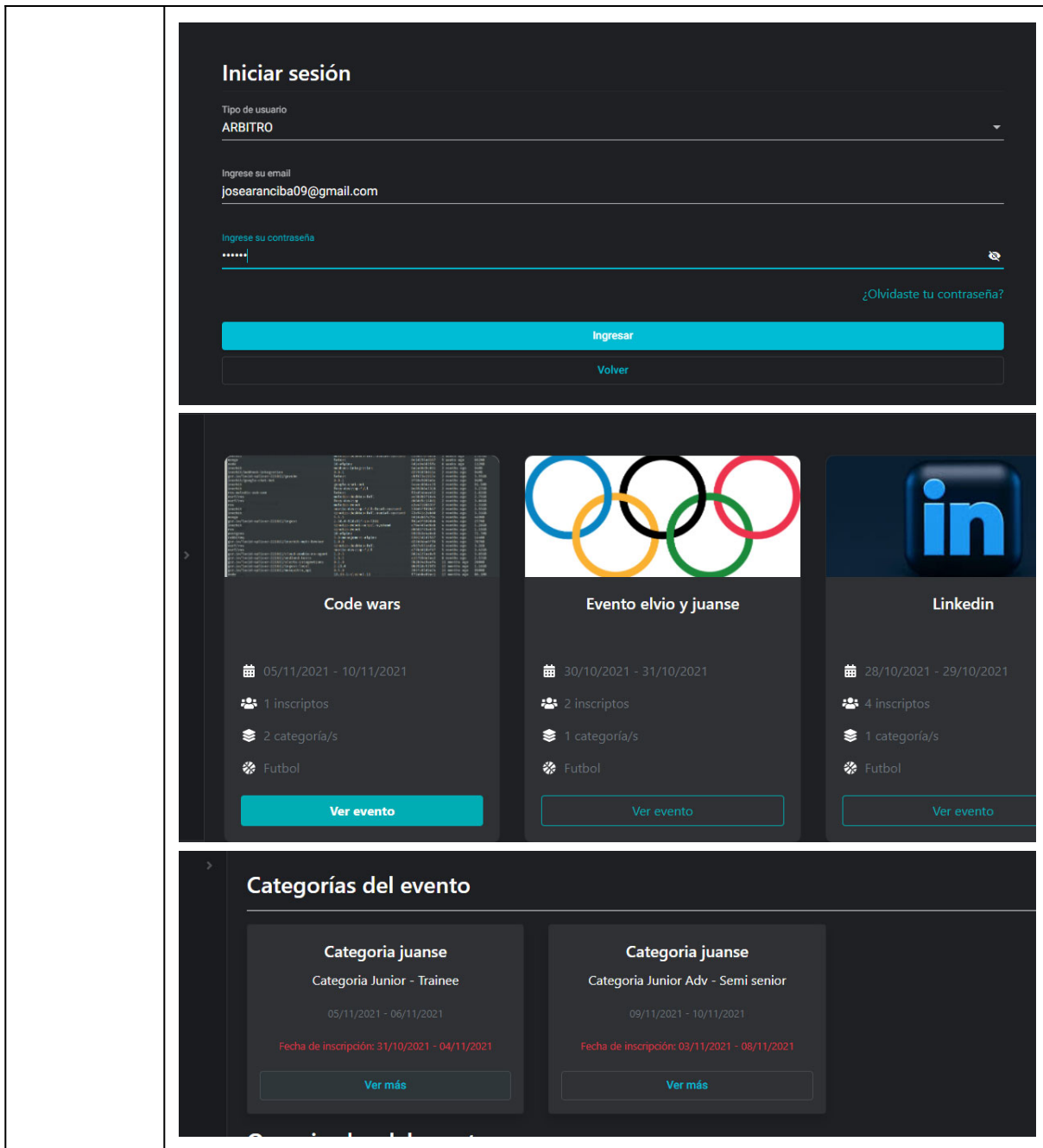
	<p>Selecciona Deportista</p> <p>Clickea en el campo mail</p> <p>Ingresa el mail :josearanciba09@gmail.com</p> <p>Clickea el campo contraseña e ingresa la contraseña : 123456 Clickea en "Ingresar"</p>	<p>campo indicando la selección correcta y habilita los campos cuil y apellido</p> <p>Se muestra un ejemplo del formato dentro del campo</p> <p>Se muestra el color azul en el campo indicando el formato correcto</p>
<p>Resultado esperado</p>	<p>El sistema muestra un mensaje de error : "el usuario no cuenta con el rol indicado"</p>	
<p>Evidencia:</p>		
<p><b>ESTADO</b></p>	<p><b>FAIL</b></p>	
<p><b>Nota:</b></p>	<p>El mensaje de error es incorrecto, debería ser : "el usuario no cuenta con el rol indicado"</p>	
<p>Acciones</p>	<p>Realizar fix en el script de manejo de errores de la página de login con roles incorrectos.</p>	

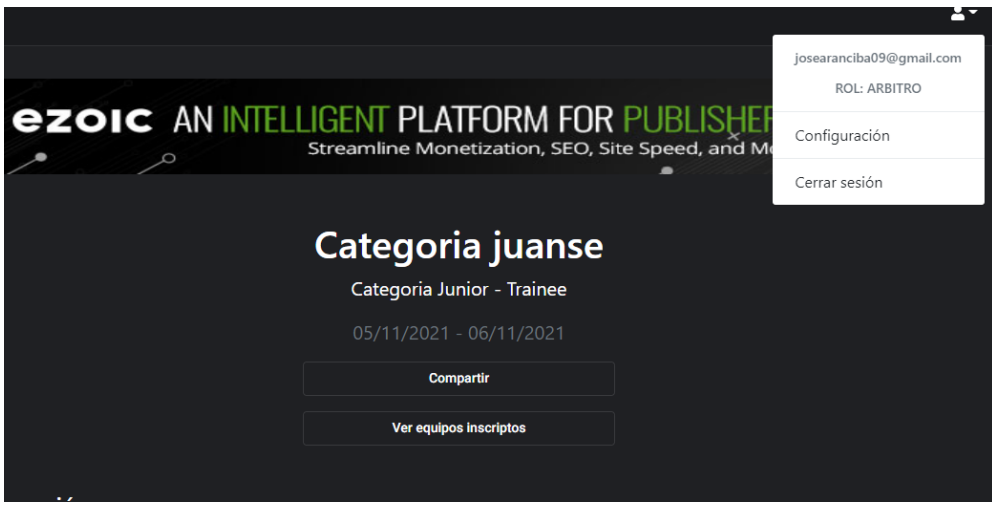
ID	TCS-02	
Nombre	Registrar usuario patrocinador enviando datos no pertenecientes al rol	
Historia de usuario	Log in	
Descripción	El usuario intenta loguearse con un rol no habilitado para ese perfil	
Precondiciones	Usuario en página de registro, usuario creado con rol patrocinador	
Datos de prueba	<pre>"nombre": "itachi", "CUIT": "2040779581", "teléfono": "02616510467", "tipoOrganizador": "IDGRATUITO", "correo": "rakexeh652@mom2kid.com", "password": "Konoha", "localidad": "IDSan Jose", "domicilio": "Aldea de la hoja 1523", "latitud": 2312312312.123812738127, "longitud": 2312312312.123812738122, "rol": "PATROCINADOR"</pre>	
Fecha Ejecución	28/08/2021	
tipo	Negativa	
Pasos	Usuario	Sistema
	Se envia la peticion con los datos de diferentes roles descriptos debajo	
Resultado esperado	Carga los datos de patrocinador e ignora los datos extras	
Dirección:	<a href="https://sportinev-back-starkloff.cloud.okteto.net/api/auth/signup">https://sportinev-back-starkloff.cloud.okteto.net/api/auth/signup</a>	
Método	POST	
Contenido	<pre>POST data: {   "nombre": "itachi",   "CUIT": "2040779581",   "teléfono": "02616510467",   "tipoOrganizador": "IDGRATUITO",   "correo": "rakexeh652@mom2kid.com",   "password": "Konoha",   "localidad": "IDSan Jose",   "domicilio": "Aldea de la hoja 1523",   "latitud": 2312312312.123812738127,</pre>	

	<pre>"longitud": 2312312312.123812738122, "rol": "PATROCINADOR" }</pre>
Evidencia:	<p>Confirmar Cuenta</p>  <pre>{   "message": "Se ha generado un link para activar su cuenta, por favor verifique su email y haga click en el link" }</pre> <pre>_id: ObjectId("612e74d466e0dd11258aef4c") fechaInicioVig: 2021-08-31T13:16:37.749+00:00 nombre: "itachi" telefono: "02616510467" correo: "rakexeh652@mom2kid.com" password: "\$2a\$10\$vEQgDYQw6fqDmxwitFVWu./o.FxcOPTBQys2NvvFuVSKwd2fYLQbG" estados: Array   0: Object     fechaInicioVig: 2021-08-31T13:16:37.746+00:00     _id: ObjectId("612e74d466e0ddc8aa8aef50")     estado: ObjectId("6115fc6c8fd6a830f52b2c8f") roles: Array   0: Object     fechaInicioVig: 2021-08-31T13:16:37.748+00:00     _id: ObjectId("612e74d466e0dd226f8aef4e")     rol: ObjectId("6113602085887dfb2d950258")     patrocinador: ObjectId("612e74d466e0dd380f8aef53") __v: 0</pre>
ESTADO	OKAY

ID	TCS-03
Nombre	Inscribirme a un evento como árbitro
Historias de usuario	Log in como arbitro visualizar eventos Visualizar categoria de eventos
Descripción	El usuario intenta inscribirse a una categoría de un evento con el rol de árbitro, no habilitado para ese perfil

Precondiciones	Usuario árbitro logueado en el sistema Usuario se encuentra en la página de login	
Respuesta esperada:	Pantalla de la categoría sin permitirle al usuario visualizar el botón de inscribirse a la categoría del evento	
Datos de prueba	"correo": " <a href="mailto:josearanciba09@gmail.com">josearanciba09@gmail.com</a> ", "contraseña": "123456" "tipo rol" : "ARBITRO"	
Fecha Ejecución	28/08/2021	
tipo	Positiva	
Pasos	Usuario	Sistema
	<p>El usuario ingresa los datos de su cuenta: "correo": <a href="mailto:josearanciba09@gmail.com">josearanciba09@gmail.com</a>, "contraseña": "123456" "tipo rol" : "ARBITRO" y clickea el boton de loguearse</p> <p>El usuario clickea en el botón de ver más de un evento</p> <p>Clickea en el botón de ver categoría en la sección de categorías de la página del evento</p>	<p>Loguea exitosamente al usuario con los permisos del árbitro y le muestra la pantalla de visualización de eventos</p> <p>Redirige al usuario a la página del evento en cuestión</p>
Resultado esperado	el sistema redirige al usuario a la pantalla de la categoría del evento sin renderizar el botón de inscripción	
Evidencia:		



	
<b>ESTADO</b>	<b>OKAY</b>

ID	TCS-04	
Nombre	Loguearme como supervisor de deportes y tener los permisos adecuados	
Historias de usuario	Log in con un rol especial	
Descripción	El usuario intenta loguearse como supervisor de deportes y poseer las acciones adecuadas según su nivel de permisos	
Precondiciones	Usuario logueado en el sistema como supervisor de deportes Usuario se encuentra en la página de login y puede ver el sidebar Rol de supervisor de deportes debe tener los permisos de : -Ver deportes -Crear deportes	
Respuesta esperada:	Mostrar un sidebar en el cual solamente se muestre las acciones permitidas para ese tipo de rol: -Ver deportes -Crear deportes	
Datos de prueba	"correo": " <a href="mailto:rierajuanse@gmail.com">rierajuanse@gmail.com</a> ", "contraseña": "123456" "tipo rol" : "Supervisor de deportes"	
Fecha Ejecución	02/11/2021	
Tipo	Positiva	
Pasos	Usuario	Sistema



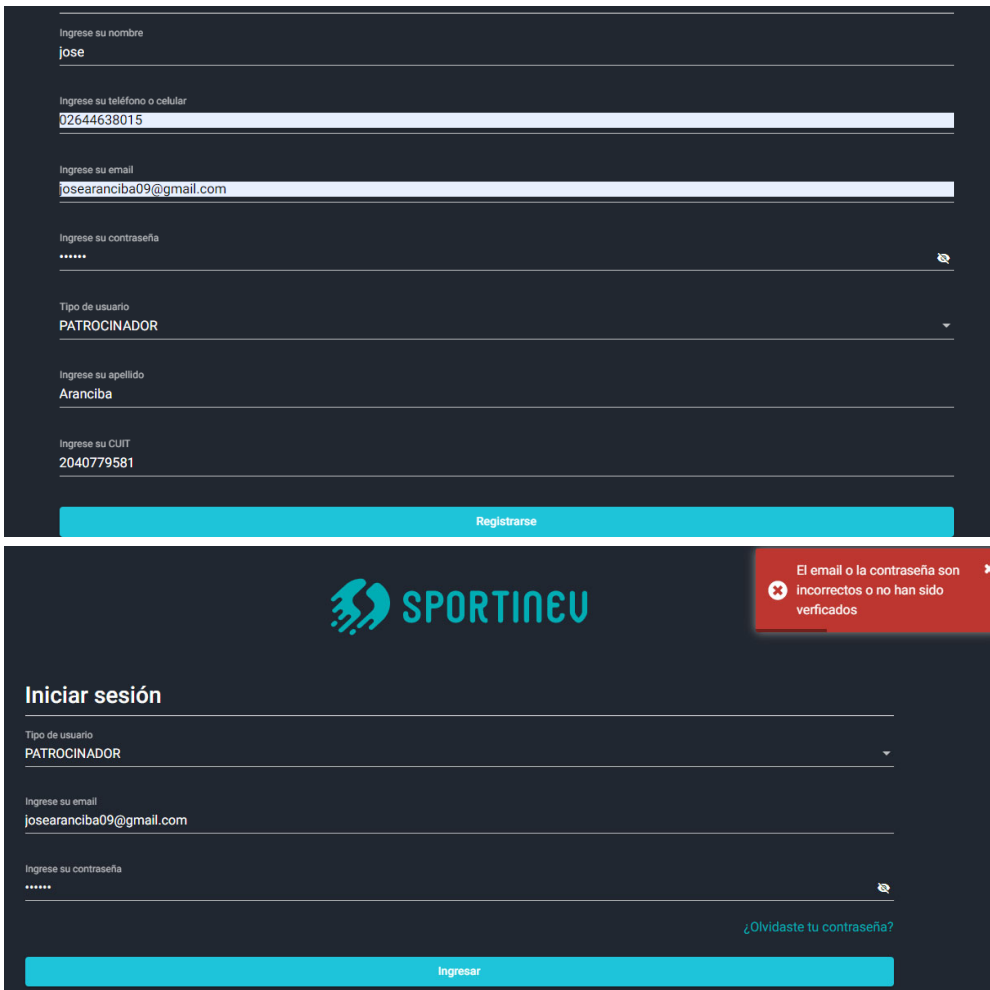
	<p>El usuario ingresa los datos de su cuenta:          "correo":  <a href="mailto:rierajuanse@gmail.com">rierajuanse@gmail.com</a>,          "contraseña": "123456"          "tipo rol" : "SUPERVISOR DE DEPORTES"          y clickea el boton de loguearse</p> <p>El usuario clickea en el botón del sidebar</p>	<p>Loguea exitosamente al usuario con los permisos del supervisor de deportes y le muestra la pantalla de visualización de eventos</p> <p>Muestra al usuario las acciones con los permisos correspondientes a este usuario:          -Ver deportes          -Crear deportes</p>
<p>Evidencia:</p>		
<p><b>ESTADO</b></p>	<p><b>FAIL</b></p>	

NOTA	El sistema muestra el permiso de ver backup, cuando el usuario no tiene el mismo asignado.
Acciones:	Realizar un fix de los guards en el front end, verificando que los permisos surjan efecto como deberían para los roles especiales y para el permiso de backup.

## Pruebas de Integración

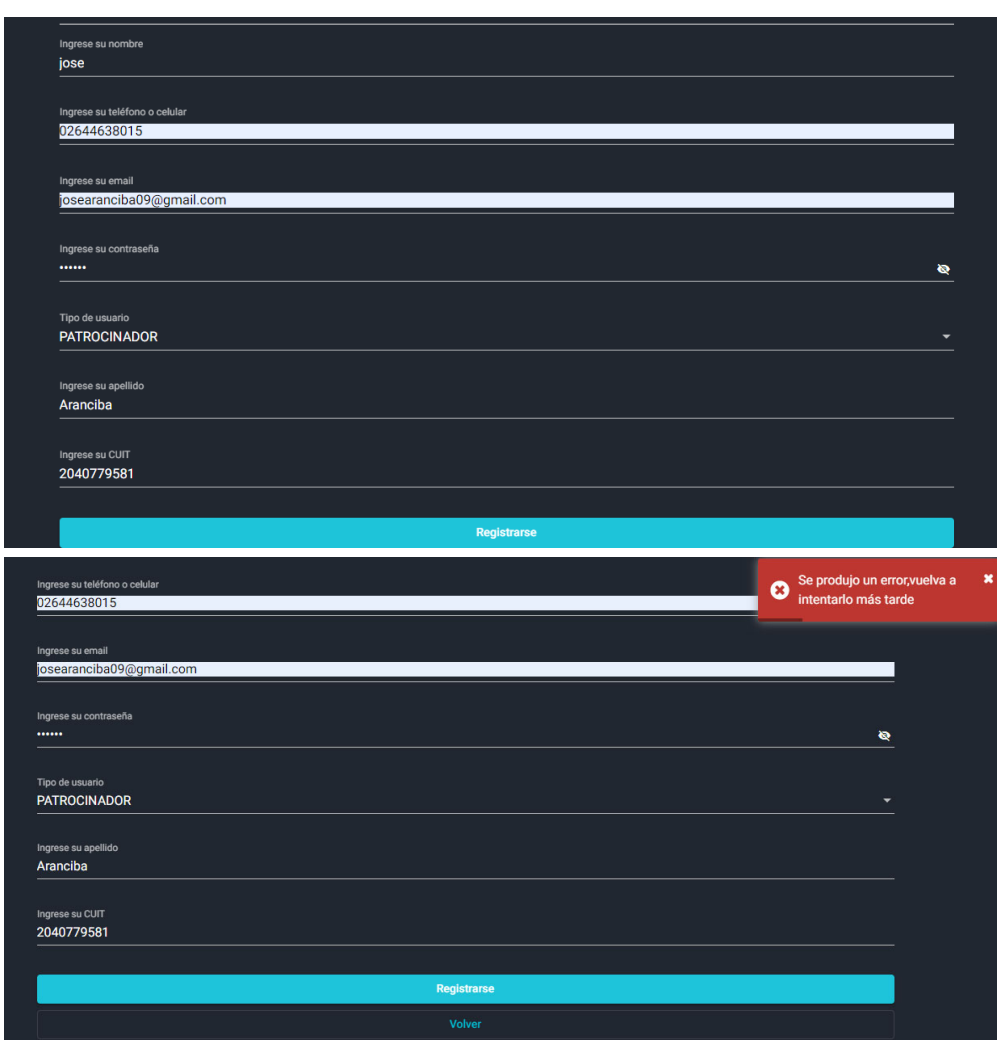
ID	TCI-01	
Nombre	Registrarse como usuario nuevo y loguearse sin validar por mail	
Historia de usuario	Registrar Usuario	
Descripción	El usuario se registra e intenta loguearse sin validación	
Precondiciones	Usuario en página de registro	
Datos de prueba	mail: josearanciba09@gmail.com contraseña: 123456 cuil:40779581 nombre: Jose Apellido: Aranciba Tipo de Usuario: Patrocinador	
Fecha Ejecución	28/08/2021	
tipo	Negativa	
Pasos	Usuario	Sistema
	Clickea en el combobox tipo usuario  Selecciona Patrocinador  Clickea en el campo mail  Ingresa el mail :josearanciba09@gmail.com	Se muestran los tipo usuarios existentes y habilitados  Se muestra el color azul en el campo indicando la selección correcta y habilita los campos cuil y apellido  Se muestra un ejemplo del formato dentro del campo

	<p>Clickea el campo contraseña e ingresa la contraseña : 123456</p> <p>Clickea en el campo nombre e ingresa José</p> <p>Clickea en el campo apellido e ingresa Aranciba</p> <p>Clickea en el campo cuil e ingresa 20407795815</p> <p>Clickea en "Registrarse"</p> <p>Clickea en el combobox tipo usuario</p> <p>Selecciona Patrocinador</p> <p>Clickea en el campo mail</p> <p>Ingresa el mail :josearanciba09@gmail.com</p> <p>Clickea el campo contraseña e ingresa la contraseña : 123456</p> <p>Clickea en ingresar</p>	<p>Se muestra el color azul en el campo indicando el formato correcto</p> <p>El sistema devuelve al usuario a la página de login.</p> <p>Se muestran los tipo usuarios existentes y habilitados</p> <p>Se muestra el color azul en el campo indicando la selección correcta</p> <p>Se muestra un ejemplo del formato dentro del campo</p> <p>Se muestra el color azul en el campo indicando el formato correcto</p>
<p>Resultado esperado</p>	<p>El sistema muestra un mensaje de error al no haber validado el usuario</p>	

<p>Evidencia:</p>	
<p><b>ESTADO</b></p>	<p><b>OKAY</b></p>

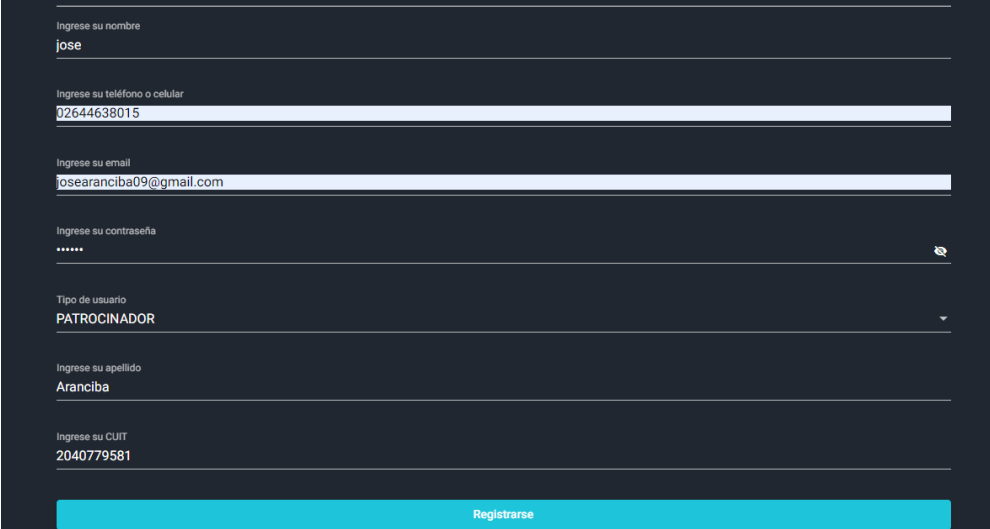
<p>ID</p>	<p>TCI-02</p>
<p>Nombre</p>	<p>Registrarse como usuario previamente creado</p>
<p>Historia de usuario</p>	<p>Registrar Usuario</p>
<p>Descripción</p>	<p>El usuario se registra con un usuario existente</p>
<p>Precondiciones</p>	<p>Usuario en página de registro</p>
<p>Datos de prueba</p>	<p>           mail: josearanciba09@gmail.com            contraseña: 123456            cuil:40779581            nombre: Jose            Apellido: Aranciba            Tipo de Usuario: Patrocinador         </p>

Fecha Ejecución	28/08/2021	
tipo	Negativa	
Pasos	Usuario	Sistema
	<p>Clickea en el combobox tipo usuario</p> <p>Selecciona Patrocinador</p> <p>Clickea en el campo mail</p> <p>Ingresa el mail :josearanciba09@gmail.com</p> <p>Clickea el campo contraseña e ingresa la contraseña : 123456</p> <p>Clickea en el campo nombre e ingresa José</p> <p>Clickea en el campo apellido e ingresa Aranciba</p> <p>Clickea en el campo cuil e ingresa 20407795815</p> <p>Clickea en "Registrarse"</p>	<p>Se muestran los tipo usuarios existentes y habilitados</p> <p>Se muestra el color azul en el campo indicando la selección correcta y habilita los campos cuil y apellido</p> <p>Se muestra un ejemplo del formato dentro del campo</p> <p>Se muestra el color azul en el campo indicando el formato correcto</p>
Resultado esperado	El sistema muestra un mensaje de error al ya haber un usuario registrado con ese mail	




<p>Evidencia:</p>	
<p><b>ESTADO</b></p>	<p><b>OKAY</b></p>

<p>ID</p>	<p>TCI-03</p>
<p>Nombre</p>	<p>Registrarse como usuario patrocinador y validar: ingresar correctamente y visualizar eventos</p>
<p>Historia de usuario</p>	<p>login</p>
<p>Descripción</p>	<p>El usuario se registra e intenta loguearse con confirmación, luego accede a la visualización de eventos</p>
<p>Precondiciones</p>	<p>Usuario en página de registro</p>
<p>Datos de prueba</p>	<p>mail: josearanciba09@gmail.com contraseña: 123456</p>

	Tipo de Usuario: Patrocinador	
Fecha Ejecución	28/08/2021	
tipo	Positiva	
Pasos	Usuario	Sistema
	<p>Clickea en el combobox tipo usuario</p> <p>Selecciona Patrocinador</p> <p>Clickea en el campo mail</p> <p>Ingresa el mail :josearanciba09@gmail.com</p> <p>Clickea el campo contraseña e ingresa la contraseña : 123456</p> <p>Clickea en el campo nombre e ingresa José</p> <p>Clickea en el campo apellido e ingresa Aranciba</p> <p>Clickea en el campo cuil e ingresa 20407795815</p> <p>Clickea en "Registrarse"</p> <p>Ingresa al mail y selecciona el mail de validación</p> <p>Clickea en el link del mail</p> <p>El usuario se dirige a la página de login</p> <p>Clickea en el combobox tipo usuario</p>	<p>Se muestran los tipos de usuarios existentes y habilitados</p> <p>Se muestra el color azul en el campo indicando la selección correcta y habilita los campos cuil y apellido</p> <p>Se muestra un ejemplo del formato dentro del campo</p> <p>Se muestra el color azul en el campo indicando el formato correcto</p> <p>El sistema devuelve al usuario a la página de login.</p> <p>Se muestra el mensaje la cuenta se ha creado con éxito y habilita la cuenta del usuario</p> <p>Se muestran los tipos de usuarios existentes y habilitados</p> <p>Se muestra el color azul en el campo indicando la selección</p>

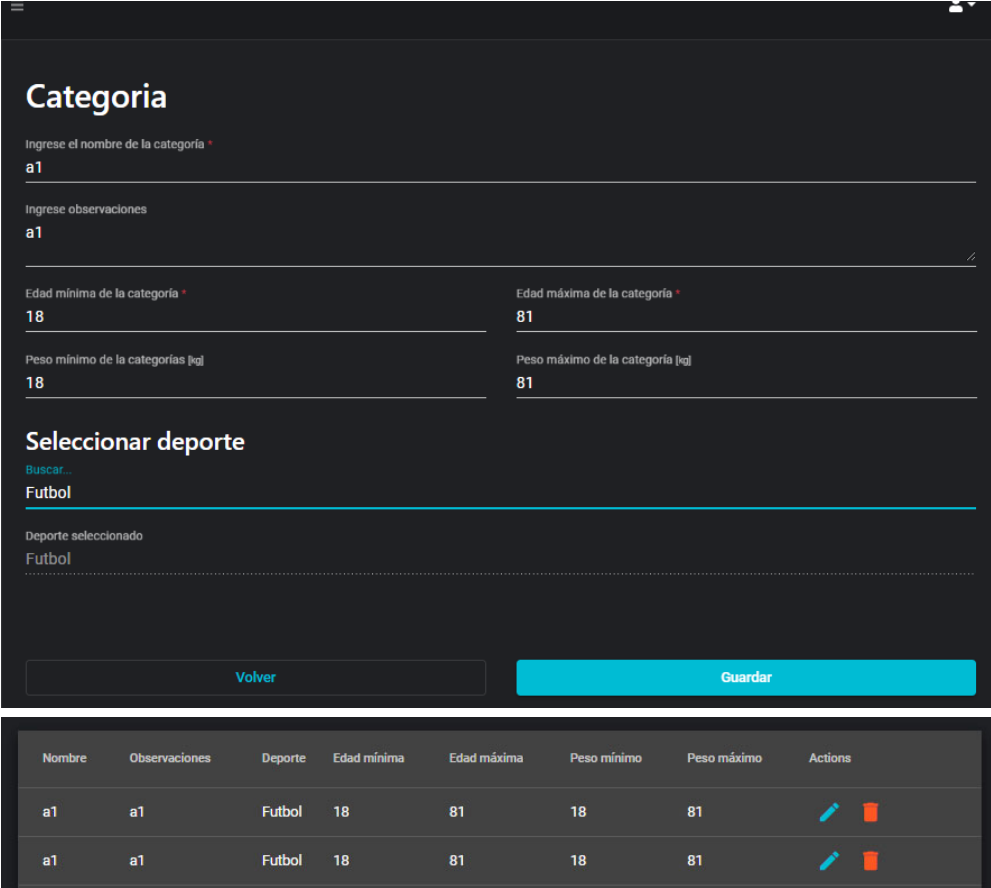












	<p>Selecciona Patrocinador</p> <p>Clickea en el campo mail</p> <p>Ingresa el mail :josearanciba09@gmail.com</p> <p>Clickea el campo contraseña e ingresa la contraseña : 123456</p> <p>Clickea en ingresar</p>	<p>correcta</p> <p>Se muestra un ejemplo del formato dentro del campo</p> <p>Se muestra el color azul en el campo indicando el formato correcto</p>
<p>Resultado esperado</p>	<p>El sistema muestra la pantalla de visualización de eventos correctamente</p>	
<p>Evidencia:</p>		



	<div data-bbox="459 322 1299 685" style="border: 1px solid black; padding: 10px; text-align: center;"> <h3>Confirmar Cuenta</h3> <p>Hola jose</p> <p>Has solicitado dar de alta tu Cuenta.</p> <p>Has click en el link que esta debajo para Confirmar tu Cuenta</p> <p><a href="#">Confirmar mi cuenta</a></p> <p>Ignora este mail si no has solicitado esta operación</p> </div> <div data-bbox="395 719 1394 1126" style="background-color: #1a2b3c; color: white; padding: 20px; text-align: center;">  <h2>La cuenta se ha creado con éxito</h2> </div> <div data-bbox="395 1167 1394 1823" style="background-color: #1a2b3c; color: white; padding: 20px;"> <h3>Eventos</h3> <p>Buscar evento <span style="float: right;">▼</span></p> <div style="display: flex; justify-content: space-around;"> <div data-bbox="475 1424 887 1823" style="background-color: #2c3e50; color: white; padding: 10px; text-align: center;">  <p><b>Evento, aguante Sportinev!</b></p> <p>📅 04/11/2021 - 05/11/2021</p> <p>👤 4 inscriptos</p> </div> <div data-bbox="927 1424 1339 1823" style="background-color: #2c3e50; color: white; padding: 10px; text-align: center;">  <p><b>PRUEBA NACHO</b></p> <p>📅 05/11/2021 - 06/11/2021</p> <p>👤 1 inscriptos</p> </div> </div> </div>
ESTADO	OKAY

ID	TCI-04
----	--------

Nombre	Loguearse como organizador y crear dos categorías con mismo nombre y obtener error	
Historia de usuario	Crear Categoría	
Descripción	El usuario organizador o administrador se registra e intenta crear 2 categorías con el mismo nombre	
Precondiciones	Usuario con cuenta organizador existente	
Datos de prueba	rol: ORGANIZADOR email: <a href="mailto:josearanciba09@gmail.com">josearanciba09@gmail.com</a> contraseña:123456 Categoría[ nombre: {a1} observaciones: {a1} edad mínima {18} edad máxima {81} peso mínimo {18} peso máximo {81} ]	
Fecha Ejecución	28/08/2021	
tipo	Positiva	
Pasos	Usuario	Sistema
	<p>El usuario organizador se loguea con los datos:          rol: ORGANIZADOR          email: <a href="mailto:josearanciba09@gmail.com">josearanciba09@gmail.com</a>          contraseña:123456</p> <p>Clickea en la pestaña “Creación de categoría”</p> <p>Ingresa los datos de la categoría de la siguiente forma:          nombre: {a1}          observaciones: {a1}          edad mínima {18}          edad máxima {81}          peso mínimo {18}          peso máximo {81}          y clickea en guardar</p>	<p>Muestra los eventos y el sidebar con la pestaña de creación de categorías</p> <p>Muestra el formulario con los datos de categoría a ingresar</p> <p>Crea correctamente la categoría</p>

	<p>Ingresar nuevamente a la pantalla de creación de categoría e ingresar los mismos datos de nombre nuevamente:</p> <p>       nombre: {a1}        observaciones: {a1}        edad mínima {18}        edad máxima {81}        peso mínimo {18}        peso máximo {81}        y clickea en guardar     </p>																									
<p>Resultado esperado</p>	<p>el sistema muestra el error: Ya existe una categoría con el mismo nombre</p>																									
<p>Evidencia:</p>	 <table border="1" data-bbox="395 1758 1391 1926"> <thead> <tr> <th>Nombre</th> <th>Observaciones</th> <th>Deporte</th> <th>Edad mínima</th> <th>Edad máxima</th> <th>Peso mínimo</th> <th>Peso máximo</th> <th>Actions</th> </tr> </thead> <tbody> <tr> <td>a1</td> <td>a1</td> <td>Futbol</td> <td>18</td> <td>81</td> <td>18</td> <td>81</td> <td> </td> </tr> <tr> <td>a1</td> <td>a1</td> <td>Futbol</td> <td>18</td> <td>81</td> <td>18</td> <td>81</td> <td> </td> </tr> </tbody> </table>		Nombre	Observaciones	Deporte	Edad mínima	Edad máxima	Peso mínimo	Peso máximo	Actions	a1	a1	Futbol	18	81	18	81	 	a1	a1	Futbol	18	81	18	81	 
Nombre	Observaciones	Deporte	Edad mínima	Edad máxima	Peso mínimo	Peso máximo	Actions																			
a1	a1	Futbol	18	81	18	81	 																			
a1	a1	Futbol	18	81	18	81	 																			

<b>ESTADO</b>	<b>FAIL</b>
Actividades	Realizar la programación de las funciones de validación de unicidad de entidades respecto a categorías

Prueba de integración entre módulos seguridad y sharing

ID	TCI-05	
Nombre	Creación de un equipo y luego compartirlo en twitter	
Historia de usuario	Alta equipo compartir evento	
Descripción	El usuario deportista crea un evento y luego de ser creado, lo comparte en twitter exitosamente	
Precondiciones	Usuario en página de creación de equipo logueado como deportista Usuario integrante habilitado Usuario con permisos para compartir equipos	
Datos de prueba	nombre: San Lorenzo descripción: <3<3<3 integrantes: starkloffnacho@gmail.com	
Fecha Ejecución	28/08/2021	
tipo	Negativa	
Pasos	Usuario	Sistema
	<p>Completa el formulario de creación de equipo con los datos: nombre: San Lorenzo descripción: &lt;3&lt;3&lt;3 integrantes: <a href="mailto:starkloffnacho@gmail.com">starkloffnacho@gmail.com</a> y clickea en el botón de guardar</p> <p>El usuario clickea en el símbolo de compartir equipo en la misma fila del equipo que creó recientemente</p>	<p>Guarda el equipo y redirige al usuario en la pantalla de visualización de equipos mostrando el equipo recientemente creado</p> <p>Muestra un modal con las redes</p>

	<p>El usuario selecciona el icono de la red social twitter para poder compartir el equipo.</p>	<p>sociales posibles a compartir incluidas twitter</p>
<p>Resultado esperado</p>	<p>El sistema luego de completar los pasos, redirige al usuario a twitter con un mensaje escrito para compartir en esa plataforma el equipo recientemente generado.</p>	
<p>Evidencia:</p>		

	
ESTADO	OKAY

## Pruebas de Carga

Las pruebas de carga son la práctica de simular el uso de la aplicación como si un usuario normal lo estuviera usando para analizar e identificar factores como la capacidad de respuesta, degradación y escalabilidad.

ID	TCC-01	
nombre	Realizar peticiones de ingreso de usuario inexistente	
Historia de usuario	login	
Descripción	Se envía una petición de ingreso de un usuario que no existe devolviendo un mensaje con el error correspondiente.	
Precondiciones	Usuario en página de registro	
Datos de prueba	"correo": "starkloffnacho@gmail.com", "password": "Starkloff", "rol": "DEPORTISTA"	
Fecha Ejecución	28/08/2021	
Tipo	Negativa	
Pasos	Usuario	Sistema

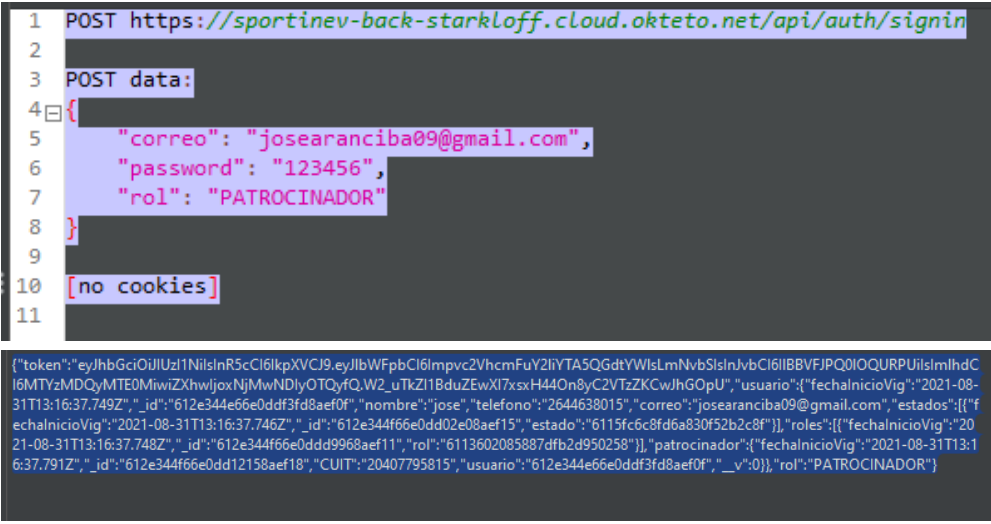
	Se envia la peticion con los datos detallados debajo	
Resultado esperado	retorna: El email o la contraseña son incorrectos	
Protocolo	HTTPS	
Método	POST	
Ruta:	sportinev-back-starkloff.cloud.okteto.net/api/auth/signin	
Evidencia:	 <pre> 1 HTTP/1.1 404 Not Found 2 Date: Tue, 31 Aug 2021 14:45:41 GMT 3 Content-Type: application/json; charset=utf-8 4 Content-Length: 102 5 Connection: keep-alive 6 X-Powered-By: Express 7 Access-Control-Allow-Origin: * 8 ETag: W/"66-1cdelpavH2DvH9AjNKJBydEYbBg" 9 Strict-Transport-Security: max-age=15724800; includeSubDomains 10 1 { 2   "correo": "starkloffnacho@gmail.com", 3   "password": "Starkloff", 4   "rol": "DEPORTISTA" 5 } </pre>	
<b>ESTADO</b>	<b>OKAY</b>	

ID	TCC-02
Nombre	Realizar peticiones de ingreso de usuario erróneas
Historia de usuario	login
Descripción	Se envía una petición de ingreso con un formato erróneo devolviendo un mensaje con el error correspondiente
Precondiciones	Usuario en página de registro
Datos de	"correo": ""

prueba		
Fecha Ejecución	28/08/2021	
Tipo	Negativa	
Pasos	Usuario	Sistema
	Se envia la peticion con los datos detallados debajo	
Resultado esperado	retorna: Error en JSON	
Protocolo	HTTPS	
Método	POST	
Ruta:	sportinev-back-starkloff.cloud.okteto.net/api/auth/signin	
Evidencia:	 <pre> 1 POST https://sportinev-back-starkloff.cloud.okteto.net/api/auth/signin 2 3 POST data: 4 { 5   "correo": "" 6 } 7 8 [no cookies] 9 ... {"status": "error", "message": "Unexpected token \r in JSON at position 18", "date": "2021-08-31T14:45:42.879Z"} </pre>	
<b>ESTADO</b>	<b>OKAY</b>	

ID	TCC-03
Nombre	Realizar peticiones de ingreso de usuario en forma correcta
Historia de usuario	login

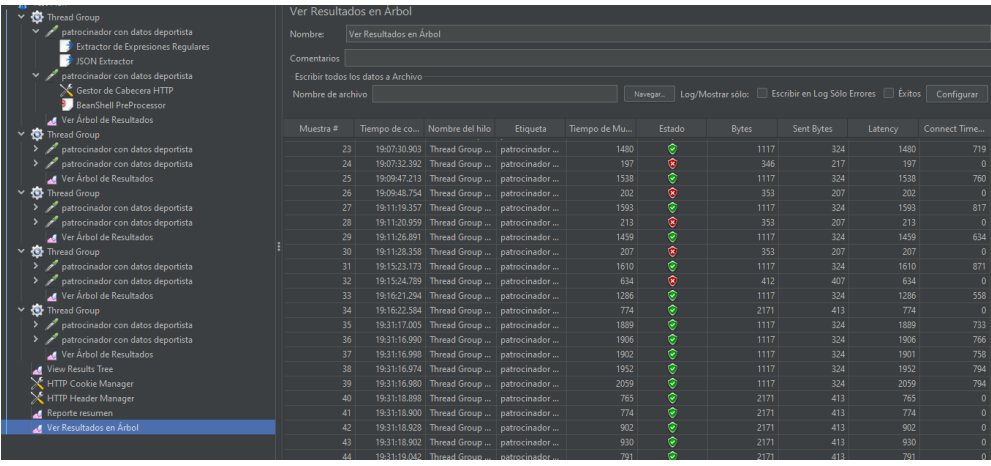


Descripción	Se envía una petición de ingreso de un usuario correcto devolviendo el token de autenticación encriptado y los datos del usuario exceptuando la contraseña.	
Precondiciones	Usuario en página de registro	
Datos de prueba	"correo": "josearanciba09@gmail.com", "password": "123456", "rol": "PATROCINADOR"	
Fecha Ejecución	28/08/2021	
Tipo	Positiva	
Pasos	Usuario	Sistema
	Se envía la petición con los datos detallados debajo	
Resultado esperado	retorna: Token de autenticación encriptado y datos del usuario logueado	
Protocolo	HTTPS	
Método	POST	
Ruta:	sportinev-back-starkloff.cloud.okteto.net/api/auth/signin	
Evidencia:	 <pre> 1 POST https://sportinev-back-starkloff.cloud.okteto.net/api/auth/signin 2 3 POST data: 4 { 5   "correo": "josearanciba09@gmail.com", 6   "password": "123456", 7   "rol": "PATROCINADOR" 8 } 9 10 [no cookies] 11  {"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFnZW50IjpbVhcmFuY2IyYTA5QGdtYWlsLmNvbSIsInVjbCI6IiBBVFJQOjQURPUIsImhhdCI6IjI021-08-31T13:16:37.749Z","id":"612e344e66e0ddf3fd8aef0f","nombre":"jose","telefono":"2644638015","correo":"josearanciba09@gmail.com","estados":{"fechalnicioVig":"2021-08-31T13:16:37.749Z","id":"612e344f66e0ddd9968aef11","estado":"6115fc6c8fd6a830f52b2c8f"},"roles":{"fechalnicioVig":"2021-08-31T13:16:37.749Z","id":"612e344f66e0ddd9968aef11","rol":"6113602085887dfb2d950258"},"patrocinador":{"fechalnicioVig":"2021-08-31T13:16:37.791Z","id":"612e344f66e0dd12158aef18","CUIT":"20407795815","usuario":"612e344e66e0ddf3fd8aef0f","_v":0},"rol":"PATROCINADOR"} </pre>	
<b>ESTADO</b>	<b>OKAY</b>	

## Pruebas de Estrés

Se realizaron pruebas de estrés, realizando múltiples consultas concurrentes al servidor en un tiempo reducido. Los tests siguientes se ejecutaron para el caso de uso de visualización de departamentos, pero en el sistema se utilizó el mismo procedimiento con las mismas cantidades de consultas e hilos para todos aquellos casos de uso en los que esté involucrado el verbo GET, en el método de la consulta http.

ID	TCE-01	
Nombre	Realizar 5 peticiones concurrentes de ingreso de usuario y consultas de visualización de departamentos	
Historia de usuario	login, visualizar departamentos	
Descripción	Se envía una petición de ingreso de un usuario correcto devolviendo el token de autenticación encriptado y los datos del usuario exceptuando la contraseña, luego esta variable es guardada y utilizada para realizar consultas a los servicios encargados de devolver información acerca de las entidades del sistema.	
Precondiciones	Usuario en página de registro	
Datos de prueba	<pre>{   "correo": "rakexeh652@mom2kid.com",   "password": "Konoha",   "rol": "PATROCINADOR" }</pre>	
Fecha Ejecución	28/08/2021	
Tipo	Positiva	
Pasos	Usuario	Sistema
	Se envia la peticion con los datos detallados debajo	
Resultado esperado	retorna: Token de autenticación encriptado y datos del usuario logueado	
Protocolo	HTTPS	

Método	POST & GET
Ruta:	POST sportinev-back-starkloff.cloud.okteto.net/api/auth/signin GET /api/departamentos/
Evidencia:	
ESTADO	OKAY
Latencia Máxima	1810
Error	0%

ID	TCE-02
Nombre	Realizar 20 repeticiones con 5 hilos concurrentes de ingreso de usuario y consultas de visualización de departamentos
Historia de usuario	login, visualizar departamentos
Descripción	Se envía una petición de ingreso de un usuario correcto devolviendo el token de autenticación encriptado y los datos del usuario exceptuando la contraseña, luego esta variable es guardada y utilizada para realizar consultas a los servicios encargados de devolver información acerca de las entidades del sistema.
Precondiciones	Usuario en página de registro
Datos de prueba	<pre>{   "correo": "rakexeh652@mom2kid.com",   "password": "Konoha", }</pre>

	"rol": "PATROCINADOR" }																																		
Fecha Ejecución	28/08/2021																																		
Tipo	Positiva																																		
Pasos	Usuario	Sistema																																	
	Se envia la peticion con los datos detallados debajo																																		
Resultado esperado	retorna: Token de autenticación encriptado y datos del usuario logueado																																		
Protocolo	HTTPS																																		
Método	POST & GET																																		
Ruta:	POST sportinev-back-starkloff.cloud.okteto.net/api/auth/signin GET /api/departamentos/																																		
Evidencia:	<table border="1"> <thead> <tr> <th>Etiqueta</th> <th># Muestras</th> <th>Media</th> <th>Min</th> <th>Máx</th> <th>Desv. Estándar</th> <th>% Error</th> <th>Rendimiento</th> <th>Kb/sec</th> <th>Sent KB/sec</th> <th>Media de By...</th> </tr> </thead> <tbody> <tr> <td>patrocinador...</td> <td>356</td> <td>984</td> <td>1</td> <td>2142</td> <td>394,70</td> <td>7,87%</td> <td>6,5/min</td> <td>0,16</td> <td>0,04</td> <td>1551,3</td> </tr> <tr> <td>Total</td> <td>356</td> <td>984</td> <td>1</td> <td>2142</td> <td>394,70</td> <td>7,87%</td> <td>6,5/min</td> <td>0,16</td> <td>0,04</td> <td>1551,3</td> </tr> </tbody> </table>		Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de By...	patrocinador...	356	984	1	2142	394,70	7,87%	6,5/min	0,16	0,04	1551,3	Total	356	984	1	2142	394,70	7,87%	6,5/min	0,16	0,04	1551,3
Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de By...																									
patrocinador...	356	984	1	2142	394,70	7,87%	6,5/min	0,16	0,04	1551,3																									
Total	356	984	1	2142	394,70	7,87%	6,5/min	0,16	0,04	1551,3																									
<b>ESTADO</b>	<b>OKAY</b>																																		
<b>Latencia Máxima</b>	<b>2142</b>																																		
<b>Error</b>	<b>7%</b>																																		

ID	TCE-03
Nombre	Realizar 20 repeticiones con 5 hilos concurrentes de ingreso de usuario y consultas de visualización de departamentos
Historia de usuario	login, visualizar departamentos
Descripción	Se envía una petición de ingreso de un usuario correcto devolviendo el token de autenticación encriptado y los datos del usuario exceptuando la contraseña, luego esta variable es guardada y utilizada para realizar consultas a los servicios encargados de devolver información acerca de las

	entidades del sistema.																																		
Precondiciones	Usuario en página de registro																																		
Datos de prueba	<pre>{   "correo": "rakexeh652@mom2kid.com",   "password": "Konoha",   "rol": "PATROCINADOR" }</pre>																																		
Fecha Ejecución	28/08/2021																																		
Tipo	Positiva																																		
Pasos	Usuario	Sistema																																	
	Se envia la peticion con los datos detallados debajo	retorna: Token de autenticación encriptado y datos del usuario logueado																																	
Resultado esperado	retorna: Token de autenticación encriptado y datos del usuario logueado																																		
Protocolo	HTTPS																																		
Método	POST & GET																																		
Ruta:	POST sportinev-back-starkloff.cloud.okteto.net/api/auth/signin GET /api/departamentos/																																		
Evidencia:	<table border="1"> <thead> <tr> <th>Etiqueta</th> <th># Muestras</th> <th>Media</th> <th>Min</th> <th>Máx</th> <th>Desv. Estándar</th> <th>% Error</th> <th>Rendimiento</th> <th>Kb/sec</th> <th>Sent KB/sec</th> <th>Media de Byt...</th> </tr> </thead> <tbody> <tr> <td>patrocinador...</td> <td>306</td> <td>983</td> <td>1</td> <td>2142</td> <td>406,64</td> <td>9,15%</td> <td>5,9/min</td> <td>0,15</td> <td>0,04</td> <td>1536,2</td> </tr> <tr> <td>Total</td> <td>306</td> <td>983</td> <td>1</td> <td>2142</td> <td>406,64</td> <td>9,15%</td> <td>5,9/min</td> <td>0,15</td> <td>0,04</td> <td>1536,2</td> </tr> </tbody> </table>		Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Byt...	patrocinador...	306	983	1	2142	406,64	9,15%	5,9/min	0,15	0,04	1536,2	Total	306	983	1	2142	406,64	9,15%	5,9/min	0,15	0,04	1536,2
Etiqueta	# Muestras	Media	Min	Máx	Desv. Estándar	% Error	Rendimiento	Kb/sec	Sent KB/sec	Media de Byt...																									
patrocinador...	306	983	1	2142	406,64	9,15%	5,9/min	0,15	0,04	1536,2																									
Total	306	983	1	2142	406,64	9,15%	5,9/min	0,15	0,04	1536,2																									
<b>ESTADO</b>	<b>ADVERTENCIA</b>																																		
<b>Latencia Máxima</b>	<b>2142</b>																																		
<b>Error</b>	<b>9%</b>																																		

Al terminar las pruebas de carga y estrés se concluye que el sistema funciona dentro de los parámetros aceptables de rendimiento para la carga diaria utilizada por el momento.

## Manual de usuarios

### ANEXO 4: Manual de usuario de Sportinev

#### Guionado de videos de manual de usuario

##### Landing Page

Una vez que se ingresa a la URL de la aplicación, la primera página con la que se encuentra el usuario es la landing page, donde se observan las características principales de Sportinev, como por ejemplo, una interfaz responsiva, fácil de usar y completamente moderna.

Otra característica del sistema es la posibilidad de ver los eventos que se encuentran en curso, sin tener que estar registrado en el sistema. Para poder visualizar los mismos, el usuario invitado deberá clicar en la pestaña llamada “eventos” en la parte superior derecha, pudiendo acceder a las características generales de los eventos del sistema como la fecha de inicio y de finalización, la cantidad de inscriptos y el tipo de deporte del evento.

La visualización de eventos, sin estar registrado, está especialmente pensada para fomentar la participación sobre los eventos del sistema y fomentar el uso de Sportinev.

##### Acceder al sistema

###### **Log In:**

Para acceder al sistema, en la página de landing, el usuario deberá clicar el botón iniciar sesión, y luego, en el formulario que aparece en la pantalla, seleccionar el tipo de rol, e ingresar su email y su contraseña.

Una vez completados todos los campos, debe presionar en el botón “ingresar”, el cual si todos los datos son válidos y correctos, le permitirá el acceso al sistema.

###### **Sign Up:**

En caso de que el usuario desee crear su cuenta, en la pantalla de login, deberá clicar el botón “registrarse” al final de la página, y luego, en el formulario siguiente, ingresar los datos básicos: nombre, teléfono, email, contraseña, y tipo de usuario

Luego, según el tipo de usuario elegido, nuevos campos deberán ser completados, por ejemplo, en caso de elegir deportista, el usuario deberá ingresar su apellido, fecha de nacimiento, DNI, país, provincia y departamento al que pertenece.

Una vez ingresados los datos correspondientes, la información se envía al clicar el botón “Registrarse” al final de la página.

Finalmente, deberá confirmar su cuenta, accediendo al email que uso a la hora de crearla.

### Cerrar Sesión

Una vez que el usuario que había iniciado sesión en la aplicación ha realizado todas las operaciones que deseaba realizar, podrá proceder a finalizar la sesión. Para lo cual deberá presionar el botón que se encuentra arriba a la derecha, esto desplegará un nuevo menú con varias opciones y en este deberá presionar el botón “Cerrar sesión”.

### Contactarse con el administrador

Cualquier usuario puede contactar al administrador del sistema para informar de errores que se han producido al utilizar la aplicación o para pedir nuevas funcionalidades. Una vez enviado el mensaje, el usuario tendrá que esperar 4 horas para poder volver a enviar otro, esto se hace a fin de evitar spam.

Para hacerlo, el usuario debe, en la página de landing, clicar en el botón “Contactanos” en la esquina superior derecha, esto lo redirigirá a un formulario que deberá llenar con datos válidos.

### Eventos

Tanto los usuarios con cuenta en el sistema, como los que no tiene una, tiene la posibilidad de ver los eventos que se encuentran cargados en el sistema. En el caso de que el usuario no tenga cuenta, deberá presionar el botón “Eventos” que se encuentra en la landing page y se lo redirigirá a una pantalla donde podrá observar la información del evento de forma parcial. En el caso de tener cuenta lo único que se debe hacer es iniciar sesión y el sistema mostrara la página de eventos.

### Visualizar eventos

#### **Buscar:**

Tanto los usuarios que tienen cuenta, como los que no, pueden buscar un evento específico por el nombre del mismo. Para esto, se debe ingresar en el campo “Buscar Eventos”, el criterio de búsqueda, esto provocará que se busquen solo los eventos que coinciden con el mismo.

#### **Filtrar:**

Alternativamente, el usuario puede filtrar los eventos según el deporte, o el estado en el que se encuentra el evento en ese momento, para esto, debe clicar el botón con símbolo de filtrado, y luego seleccionar las opciones deseadas en los campos deporte y estado.

#### **Acciones En Eventos:**

### **Inscribirse**

Un usuario que ha iniciado sesión en el sistema puede inscribirse a un evento determinado. Para esto, es necesario clicar en el evento deseado, lo que redirige al usuario a la pantalla del evento donde verá el botón “Inscribirse” en medio de la foto de portada del evento.

El usuario debe clicar en este botón, luego elegir el método de pago y el equipo que va a registrar en el evento.

### **Compartir:**

Debajo del botón para inscribirse en un evento, se encuentra el botón “Compartir”, lo que permite al usuario enviar el enlace al evento a través de su red social favorita. Actualmente, Sportinev se integra con WhatsApp, Twitter y Facebook, pero también da la posibilidad de copiar el enlace del evento al portapapeles y que se pueda compartir por cualquier red social.

### **Navegabilidad en eventos:**

En la misma pantalla de cada evento, debajo de la foto de portada, se encuentran las opciones de visualizar las categorías del evento y el organizador del evento. Para esto, el usuario solo debe clicar en los botones “ver más” para ver la categoría o “ver perfil” en caso de querer visualizar el perfil del organizador del evento.

## Inscripción a eventos

### **Pantalla de inscripción a evento:**

Para inscribirse a un evento, el deportista debe ubicarse en la pantalla inicial del sistema, analizar los diferentes eventos de la plataforma y cuando encuentre uno de su interés deberá presionar el botón “Ver más”. Esto le mostrará al usuario toda la información del evento junto con las categorías del mismo. En el caso que le interese unirse al evento, deberá entrar en la categoría en la cual desea inscribirse. Al entrar en una categoría, se podrá observar toda la información de la misma. Para poder inscribirse a la categoría, el usuario deberá clicar en el botón “Inscribirse”. Esto cargará un formulario que deberá completar con información válida. En el mismo se le pedirá al deportista que seleccione el método de pago y el equipo que desea inscribir en el evento. Una vez que el deportista ha llenado el formulario debe seleccionar el botón enviar para finalizar la inscripción, luego de esto la inscripción quedará en estado no pago hasta que el deportista pague la inscripción y el pago sea registrado.

## Perfiles

Los perfiles de los usuarios pueden ser vistos por todos los miembros de Sportinev, sin importar el rol con el que se cuente. Esto permite que los usuarios del sistema interactúen entre sí y se incentiven entre ellos a participar de los eventos. En el perfil se muestra solo la información básica de cada usuario junto con las redes sociales del mismo, en el caso que las haya cargado.



### **Feedback:**

Para realizar un feedback a un perfil, lo único que debe hacer el usuario es ingresar al evento en el que ya participó, ingresar en la categoría en la que participó, y realizar un feedback presionando el botón de feedback. Esto abrirá un formulario que el usuario deberá llenar con datos válidos. Deberá llenar el formulario con un puntaje y comentario para el evento y un puntaje y comentario para el club en el que se participó.

### Editando el perfil del usuario

Una vez que el usuario ha iniciado correctamente sesión, en la parte superior de la pantalla, deberá presionar un icono con forma de persona y se le abrirá un menú desplegable, en el cual deberá presionar el botón “Configuración”. Eso lo redirigirá a una pantalla donde podrá visualizar más información útil de acuerdo al rol con el que ingreso al sistema.

### **Configuración general:**

Todos los roles tienen la posibilidad de realizar ciertas acciones básicas que le permiten a este interactuar de manera consistente con el sistema y gestionar las configuraciones de su perfil.

Estas acciones son:

### **Compartir Perfil:**

Cuando el usuario cliquee en compartir perfil, se muestra una pantalla que le permitirá al usuario compartir de manera sencilla el enlace a su perfil por alguna de las redes sociales soportadas por el sistema. En caso de que una red social no se encuentre soportada por Sportinev, siempre se puede copiar el enlace al portapapeles y compartirlo manualmente.

### **Actualizar Datos:**

El sistema permite el control total de los datos del usuario en todo momento, mediante la acción de actualizar datos. Al hacer clic sobre esta funcionalidad, el usuario ingresa a la pantalla de modificación de datos personales donde puede corregir o actualizar en caso de necesidad los datos personales de su perfil.

### **Agregar nuevo rol a usuario:**

El usuario tiene la posibilidad de agregar a su perfil un rol nuevo, lo que lo habilitaría a realizar otras actividades, adquiriendo nuevos permisos en el sistema. Esto puede lograrse cliqueando en el botón “Agregar nuevo rol a usuario”, en la sección de configuración del perfil, y luego seleccionando el rol o tipo de usuario que se necesita agregar a ese perfil. Una vez que se selecciona esto, el formulario pide al usuario los datos correspondientes y necesarios para ese nuevo rol, una vez ingresados, debe cliquear en el botón agregar, y finalmente aparece el cartel de éxito. Una vez efectuados estos pasos el usuario ya podrá iniciar sesión con su nuevo rol.

### **Eliminar rol a usuario**

De la misma manera, el usuario puede eliminar un rol que ya posea, cliqueando en el botón “Eliminar rol a usuario”, esto mostrará un formulario en el que el usuario deberá seleccionar el tipo de usuario que desea quitar de su perfil y cliquear en el botón “eliminar”.

### **Cambiar contraseña**

Si por algún motivo el usuario necesita modificar la contraseña de login, deberá seleccionar la opción correspondiente en el apartado de configuración de perfil. Esto abrirá un formulario donde deberá ingresar, la contraseña actual y la nueva. Para finalizar deberá presionar el botón enviar. La próxima vez que vaya a iniciar sesión, deberá usar la nueva contraseña.

### **Cambiar email**

De la misma manera, el usuario puede modificar el email asociado a su cuenta seleccionando la opción “Cambiar email”, ingresando su nuevo email y presionando el botón “Enviar”. Para que el cambio de email se haga efectivo, el usuario deberá confirmar su nuevo email presionando en el enlace que se le envió al email.

### **Vincular Red Social**

El usuario deberá clicar el botón “Vincular Red Social”, luego deberá completar el formulario con las redes sociales que desee vincular o modificar, y por último deberá clicar en el botón “enviar”. El usuario puede vincular redes sociales como Facebook, Instagram, Twitter y LinkedIn.

## **Configuración Deportista:**

### **Deportes Favoritos:**

El deportista puede añadir deportes favoritos a su perfil. Para esto deberá ir al apartado de configuración de su perfil y presionar el botón “Agregar deportes favoritos”. Esto abrirá un formulario donde el usuario deberá seleccionar, de una lista, los deportes que más le gusta, y luego presionar en el botón “guardar”.

### **Equipos:**

Un deportista puede interactuar con los equipos en los que se encuentra. Para esto deberá presionar el botón “Ver equipos” que se encuentra en el sidebar del lado izquierdo. Esto lo redirigirá a una pantalla donde podrá ver todos los equipos de los que forma parte y realizar acciones sobre los mismos.

## **Configuración Organizador:**

### **Vincular cuenta a mercado pago:**

Los organizadores pueden ofrecerle a los deportistas la posibilidad de pagar el evento mediante Mercado Pago. Pero primero deben tener vinculada su cuenta de Mercado Pago a su perfil de Sportinev. Para esto deberemos iniciar sesión como organizador, ir a las configuraciones del perfil y presionar el botón “Vincular cuenta de Mercado Pago”. Esto nos redirigirá a la página oficial de Mercado Pago, donde deberemos iniciar sesión y aceptar que Sportinev tenga acceso a nuestra cuenta. Una vez efectuados estos pasos, sé lo nos

devolverá a la página de Sportinev y ya podremos empezar a crear eventos que acepten como medio de pago Mercado Pago.

## Configuración Árbitro:

Funciones especiales de cada rol

### Cuentas Prémium:

Los organizadores tiene la posibilidad de elegir distintos tipos de cuentas con distintos beneficios. Para esto poder observar esto deberemos iniciar sesión como organizador y presionar el botón “Ver cuentas prémium” en el sidebar de la izquierda. Esto nos redirigirá a una pantalla donde se nos mostrarán todos los tipos de cuentas disponibles, con sus diferentes funcionalidades y precios. Una vez que hemos elegido el tipo de cuenta que deseamos, solo debemos clicar en el botón “Comprar cuenta” y realizar el pago del servicio con Mercado Pago.

### Crear una categoría

Primero deberemos dirigirnos al apartado de gestión de categorías, para esto deberemos presionar el botón “Ver categorías” del sidebar de la parte izquierda.

Para crear una categoría, deberemos presionar el botón “Agregar”. Luego se nos mostrará un formulario como el que estamos viendo y que deberemos completar con datos válidos.

### Crear un evento

Para crear un evento, como organizador deberemos clicar el botón “Crear nuevo evento” del sidebar de la izquierda. Luego de esto, se nos mostrará un formulario que deberemos completar con, el nombre del evento, una imagen de portada, una descripción, los medios de pago que se aceptarán, el deporte, y finalmente el tipo de evento haciendo referencia al método de clasificación de equipos a llevar a cabo. Una vez completados los datos, deberemos presionar el botón “siguiente”.

Se nos mostrará una pantalla para carga las categorías que tendrá el evento. En ella, deberemos cargar la fecha de inicio y de finalización de la categoría, la fecha de inicio y de finalización de la inscripción, el precio, la cantidad máxima de equipos que pueden participar, la descripción, los clubes donde se llevara a cabo y las jornadas en que se jugará. Finalmente, deberemos presionar el botón “Guardar categoría”, en el caso que queramos más categorías deberemos realizar los pasos anteriores de vuelta, caso contrario deberemos presionar el botón “Guardar”.

## Visualización de los deportes

Como podemos observar, cuando hemos iniciado sesión como administrador, podemos observar los deportes que se encuentran activos en el sistema. Se nos muestra información útil como el nombre, la descripción, la cantidad de equipos mínimos y máximos de los distintos deportes cargados en el sistema.

### Visualización de las categorías

Como podemos observar, cuando hemos iniciado sesión como organizador, podemos consultar la información de las categorías creadas, tales como el nombre, las observaciones, el deporte, la edad mínima y la edad máxima.

### Visualización de equipos

Como podemos observar, cuando hemos iniciado sesión como club, podemos consultar la información de los equipos creados, tales como el nombre, la descripción y una lista de los integrantes que forman parte del equipo.

### Visualización y creación de roles

Como podemos observar, cuando hemos iniciado sesión como administrador, podremos ver los roles que hemos creado en el sistema, para ellos deberemos presionar el botón “Ver roles” que se encuentra en el sidebar del lado izquierdo de la pantalla. Esto no redirigirá a una pantalla donde se muestra una tabla donde se nos muestra el nombre del rol, una descripción del mismo, los permisos que tiene asignado ese rol y las acciones que se pueden realizar.

Como podemos observar en el sidebar de la izquierda también tenemos la opción de “Crear nuevo rol”, la cual una vez que la cliqueamos, nos cargara un formulario que debemos completar con datos válidos.

### Visualización y creación de países

El administrador tiene la posibilidad de gestionar los países que estarán presentes en el sistema, para lo cual se dirigirá al sidebar que se encuentra en su parte izquierda y hará clic en la categoría llamada “Países”, abriendo un submenú, el cual contiene la creación y visualización de los países. Si el administrador hace clic en la opción “Ver países”, el sistema mostrará los que se encuentran actualmente activos, además de una categoría de opciones en el cual podrá modificar y borrar el país. En caso de hacer clic en el icono del lápiz (modificar país) el usuario podrá modificar el nombre del mismo.

En la misma categoría del sidebar, tenemos otro apartado más llamado “Crear nuevo país”, en el cual al hacer clic, nos pedirá que ingresemos el nombre del nuevo País, controlando que no exista un país con el nombre ingresado.

### Publicidad

El usuario administrador puede determinar la cantidad de publicidad que se mostrará en el sistema, siendo el rango de 0 a 2 y pudiendo elegir en que parte de la pantalla ubicar las mismas (superior e inferior de la página).

Para poder efectuar esta operación, el usuario deberá iniciar sesión como administrador y seleccionar el apartado “Cargar publicidad” ubicado en el sidebar que está en la izquierda

de la pantalla. En el apartado de publicidad, podrá seleccionar el sector sobre el cual quiere cargar la publicidad y la imagen correspondiente a la misma.

## Reportes

Los usuarios pueden ver diferentes reportes y estadísticas de todas las entidades, para esto el usuario debe dirigirse a la pestaña “reportes”, ubicada en el menú desplegable lateral y luego elegir el reporte que desee visualizar.

Descripción de los listados o reportes (con datos) con un ejemplo de cada uno y la descripción de cómo hay que hacer para obtenerlos.

El sistema cuenta con varios reportes que los distintos usuarios pueden observar para poder información útil que pueden utilizar a futuro.

Uno de los reportes más interesantes que puede contemplar el organizador, es la cantidad de inscriptos por mes, de todos los eventos que organizó, del año actual. El mencionado es un reporte que sirve al organizador para poder obtener estadísticas y métricas sobre en qué meses se inscriben más deportistas. De esta forma poder fomentar la realización de eventos en los meses cercanos al mismo, ya que podrá obtener una mayor participación y difusión en los mismos.



Figura 267: Reporte de la cantidad de inscripciones por mes

Otro reporte interesante es la cantidad de eventos que hay por deporte, esto permite al usuario organizador determinar cuáles son los deportes más populares en el sistema, y

poder planificar eventos futuros con base en ellos. Dicho reporte se muestra en un muy conveniente gráfico de barras.



Figura 268: Reporte de los deportes más utilizados por deporte

Finalmente otro reporte que puede visualizar el administrador del sistema es la cantidad de planes que pueden contratar los organizadores, que hay en el sistema, es decir cuántos usuarios tiene la cuenta “barata”, cuántos tienen la cuenta “gratis” y cuántos tienen la cuenta “prémium”. Esto le permitirá al administrador realizar actividades y descuentos para por ejemplo fomentar el uso de las cuentas prémium en la aplicación.

## Normas necesarias

Con la implementación del sistema Sportinev, el uso de formularios en papel ya no serán necesarios, los árbitros ya no necesitarán anotar los resultados de los partidos a mano, lo que evita confusiones respecto a los resultados. Los árbitros ahora pueden ingresar los resultados de los partidos al final de la jornada en el sistema, y este tratará los datos de una forma más segura.

## Organizadores

Una vez implementado el sistema, los organizadores, tendrán mayor facilidad para crear sus eventos a la hora de conseguir equipos y clubes que participen en este, ya que al poder

obtener contacto con estos a través del sistema, la difusión del evento con las personas interesadas se incrementa notablemente.

Al momento de la inscripción de los equipos, los organizadores y los equipos, se ven beneficiados por la implementación del módulo de pagos, que registrara y manejara las transacciones entre las partes. Guardando los datos en la base de datos, reduciendo el riesgo de pérdida de la información o carga errónea de la misma.

Una de las mayores ventajas del sistema y bienes más preciados para los usuarios, es la información disponible en él. Gracias a esta, los usuarios pueden tomar mejores decisiones respecto al ámbito de aplicación y lograr una mayor satisfacción respecto a la usabilidad del sistema. Debido a esto existe el módulo reportes, que muestra varios tipos de reportes a los diferentes roles del sistema, con basta información sobre los usuarios deportistas, clubes, patrocinadores, árbitros y organizadores que forman parte del sistema o de un evento en particular. Puede ser mostrada de formas muy dinámicas para satisfacer las expectativas del usuario. Previamente, los organizadores debían cargar esta información en hojas de cálculo y realizar un arduo trabajo matemático para mostrar los mismos gráficos y estadísticas que ofrece la aplicación en un solo clic. Ya no debe cargar manualmente estos datos y más importante aún, tiene la información de los demás eventos y organizadores a su alcance, que previamente era demasiado complicado, o en algunos casos imposible de conseguir.

### **Patrocinadores**

Con respecto a los patrocinios de los eventos, los organizadores ya no necesitan contactar exhaustivamente a los sponsors que él conoce. El sistema promueve la participación de los patrocinadores, mostrándoles la información más relevante de cada uno de los eventos, para que este pueda decidir cuál es el más adecuado para publicitar su marca.

Para esto, un patrocinador puede explorar los eventos disponibles en el sistema, y si identifica algún evento que es de su agrado, a partir de la información de reputación del organizador, los equipos y la cantidad de estos que participaran (Datos disponibles no solo a los usuarios registrados).

El patrocinador efectúa la petición de patrocinio del evento, para que luego, el organizador decida si aceptar o rechazar la propuesta, simplificando enormemente este procedimiento.

### **Árbitros**

Sportinev también facilita la forma de contratar árbitros para sus eventos.

### **Deportistas**

Los deportistas, ahora tienen un mejor modo de ver los eventos disponibles, únicamente accediendo al apartado principal del sistema, que es donde se muestran todos los eventos.

Para formar parte de un equipo, ya no necesitan llevar el registro de los equipos a los que pertenecen, ni recordar los integrantes, ya que estos datos son manejados por el sistema. Ya no necesitan conocer a las personas con las que quiera formar un equipo, ya que una vez dentro del sistema, el deportista puede explorar los diferentes usuarios que se encuentran registrados y realizar peticiones para formar equipo con ellos, si así lo desea. Creando un vínculo entre ellos y fomentando la creación de los equipos.

## Clubes

Los clubes pueden cargar, mantener y publicitar sus servicios en el sistema, ofreciendo sus instalaciones para desarrollar los deportes de los eventos, formando parte de la cadena de beneficios o economía circular que ofrece Sportinev a todos sus usuarios.

Además de permitirle un mayor alcance de los servicios que ofrece el sistema. Le permitirá llevar un registro y orden de estos lugares, los momentos en los que estos sitios están ocupados, con que eventos y quienes jugarán en ellos.

## Planificación de Implementación del Sistema

### Objetivos de la implementación del Sistema

El objetivo de la implementación del sistema es correr el mismo en la nube. La parte del cliente al ser desarrollada con formato responsive, los usuarios podrán utilizar la aplicación desde cualquier dispositivo que tenga un navegador.

El sistema que se va a implementar está pensado tanto para aquellas personas que desean participar en un evento y para aquellas que decidan organizar un evento y para los patrocinadores que deseen prestar su confianza en algún organizador.

### Personas involucradas en la implementación

En esta etapa se definen las tareas y se prepara la organización del equipo encargado de la etapa de implementación.

El equipo de trabajo estará conformado por:

- Scrum Master, encargado del equipo de implementación.
- DBA, encargado de implementar la base de datos a niveles productivos y la carga de datos inicial.
- Un DevOps encargado de crear, configurar los clústeres para los niveles productivo y desplegar la aplicación.
- Un QA encargado de probar la aplicación previo a lanzarla a producción.
- Desarrollador.



Cada miembro del equipo será responsable de las diferentes etapas y tendrá asignado una cantidad de horas para realizar dichas etapas

## Temas del Plan de Implementación

En primer lugar, es necesario calcular adecuadamente el esfuerzo y los recursos necesarios para llevar a cabo con éxito la implementación.

Por lo tanto, es necesario contemplar las siguientes tareas:

- La preparación y la creación de un clúster de Kubernetes con estándares de nivel productivos
- Asignación de DNS y configuración del TLS para producción.
- Creación y despliegue de una base de datos que soporte niveles productivos.
- Carga de datos iniciales en la base de datos.
- Creación de un usuario administrador.
- Verificación que las funcionalidades básicas funcionen adecuadamente.
- Capacitación del administrador del sistema

Estas tareas serán explicadas a lo largo de este documento.

## Instalación de servidores

Al utilizar los servicios de google cloud, ningún tipo de instalación de servidores es necesaria, ya que es un servicio cloud.

Las bases de datos estarán ejecutándose como contenedores en el GKE lo cual no hace que tengamos que tener costos adicionales para contratar un servidor de bases de datos.

## Equipos de conectividad, comunicaciones, configuración, infraestructura, configuración de servidores

Al tratarse de un sistema que está completamente en la nube, todas las configuraciones mencionadas en el título son responsabilidad de Google, ya que nosotros estamos contratando su servicio.

## Método de conversión elegido

Dado que no hay un sistema previo ya desarrollado, el método de conversión elegido es el directo. La implementación tendrá una duración de 2 días para poner el sistema en ambiente productivo.

## Conocimientos requeridos

Para utilizar apropiadamente el sistema, el usuario deberá poseer los siguientes conocimientos:

- Conocimientos básicos de computadoras, con cualquier sistema operativo
- Conocimientos de utilización de navegadores web, como lo son Google Chrome, Firefox.

## Requerimientos técnicos

Para el correcto funcionamiento del sistema, el usuario deberá poseer una computadora o celular con los siguientes requerimientos mínimos:

En el caso de que el usuario entre a la aplicación desde una computadora personal o notebook:

- 2 GB de memoria RAM.
- Intel i3-2130
- Conexión a internet.
- Navegador web moderno instalado.

En el caso de que el usuario ingrese a la aplicación desde una tablet o dispositivo móvil:

- Pantalla con resolución superior a 1280x720px.
- Pantalla mínima de 10".
- Conexión a internet inalámbrica o datos.
- Android 4.0.4 o iOS 5.

## Cronograma de la Implementación

Número	Tarea	Duración (Horas)	Predecesora	Recursos
1	Definir equipo de Implementación	2		Scrum Master
2	Preparación y creación de un clúster de Kubernetes	6		DevOps

	con estándares de nivel productivos			
3	Asignación de DNS y configuración del TLS para producción.	2	1,2	DevOps
4	Creación y despliegue de una base de datos que soporte niveles productivos.	8	3	DBA
5	Carga inicial de datos	6	4	DBA
6	Creación de un usuario administrador.	1	4	Desarrollador
7	Verificación que las funcionalidades básicas funcionen adecuadamente.	12	6	QA
8	Elaborar plan de recuperación	6	7	Scrum Master
9	Capacitación del administrador del sistema	5	7	Analista

## Preparación y creación de un clúster de Kubernetes con estándares de niveles productivos

El encargado de esta etapa es el DevOps.

El DevOps deberá preparar un ambiente que soporte los niveles de consumo productivos, esto significa que deberá preparar un ambiente que soporte la carga y la demanda de muchos usuarios interactuando con los componentes del sistema.

En esta etapa el DevOps creará en el proveedor de cloud que eligió para el despliegue un clúster de Kubernetes, con nodos de un tamaño considerable (con una buena cantidad de VCPU dedicada y RAM) y además deberá configurar los deployments para que sean auto escalables. Esto significa que a medida que el tráfico de usuarios aumente el clúster se encargará de generar más instancias de la aplicación para balancear la carga de forma adecuada y poder autogestionar la cantidad de recursos que necesita para un tráfico determinado.

Una vez que el ambiente esté levantado y funcionando con un margen de uso de recursos sostenible, se puede pasar a asignarle un DNS y la correspondiente configuración de los certificados para producción.

## Asignación de DNS y configuración del TLS para producción.

Para la configuración del TLS en producción utilizaremos Let's Encrypt.

El objetivo de Let's Encrypt y el protocolo ACME es hacer posible la configuración de un servidor HTTPS y hacer que obtenga automáticamente un certificado confiado por el

navegador, sin ninguna intervención humana. Esto se logra ejecutando un agente de manejo de certificados en un servidor de web.

Hay dos pasos para este proceso. Primero, el agente le prueba al AC que el servidor de web controla el dominio. Luego, el agente puede pedir, renovar, y revocar certificados para ese dominio.

### Validación de Dominio

Let's Encrypt identifica el administrador del servidor por llave pública. La primera vez que el software del agente interactúa con Let's Encrypt, genera un nuevo par de llaves y demuestra al Let's Encrypt CA que el servidor controla uno o más dominios. Esto es similar al proceso tradicional de un AC de crear una cuenta y agregar dominios a esa cuenta.

Para iniciar el proceso, el agente le pregunta al Let's Encrypt CA lo que hay que hacer para demostrar que controla example.com. El mismo mirará el nombre de dominio que se solicita y emitirá uno o más conjuntos de retos. Estas son diferentes maneras que el agente puede demostrar control sobre el dominio.

Para la emisión de certificados del lado interno del servidor usaremos **cert manager** para Kubernetes.

**cert-manager** agrega certificados y emisores de certificados como tipos de recursos en los clústeres de Kubernetes y simplifica el proceso de obtención, renovación y uso de esos certificados.

Asegurará que los certificados sean válidos y estén actualizados, e intentará renovar los certificados a la hora configurada antes de su vencimiento.

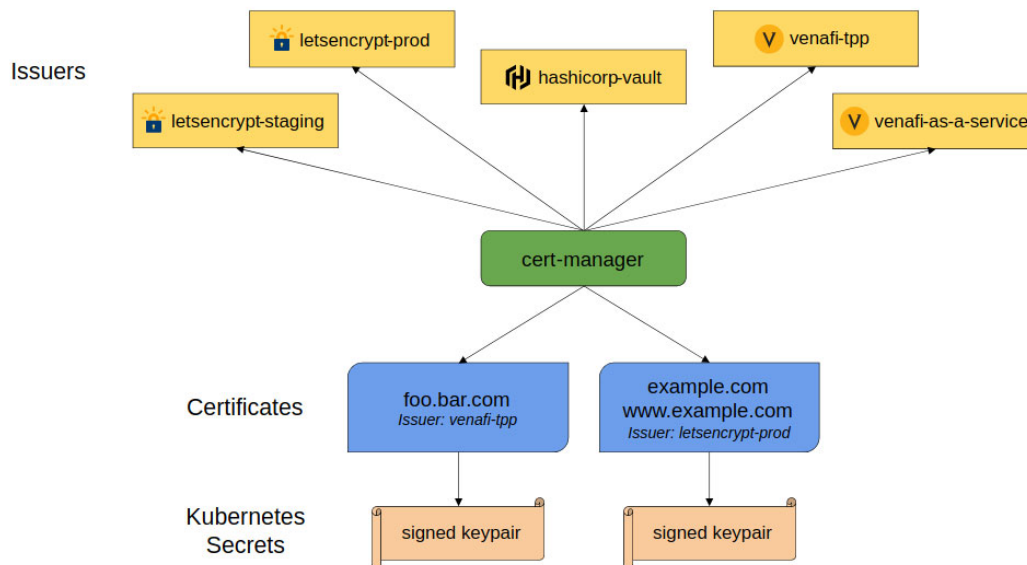


Figura 269: Funcionamiento del cert-manager

Creación y despliegue de una base de datos que soporte niveles productivos.

Una vez desplegado el clúster donde correrán las aplicaciones, se deberá desplegar la base de datos en un servidor que también se adapte a los niveles productivos, para esto utilizaremos los clústeres que ofrece Atlas MongoDB.

Se creará un clúster con 3 instancias para tener disponibilidad absoluta en todo momento y poder balancear el tráfico de usuarios a través de distintas instancias según se necesite.

## Carga inicial de datos

Una vez desplegada la base de datos en nuestro clúster de Mongo Atlas. Procederemos a la carga inicial de datos, definiendo los tipos eventos, tipos deportes, permisos, roles, provincias, departamentos, localidades, categorías y todo tipo de datos iniciales que se necesiten para que el sistema pueda realizar sus funciones correctamente.

## Migración

Debido a la inexistencia de un sistema previo, no será necesaria una migración de datos del sistema antiguo al nuevo sistema.

## Creación de un usuario administrador

Se generará un usuario administrador para poder ejecutar todos los ajustes que se necesiten desde los ABMs que tiene disponible el sistema, para que el arranque de la aplicación sea el mejor posible, se ajuste a la mejor experiencia de usuario y más completa posible.

Además, este será el usuario encargado de efectuar todas las funciones de administración del sistema y será el único usuario que será creado directamente desde la base de datos, ya que todos los otros usuarios serán creados a través del sistema.

Por este motivo, ya que este usuario necesita ser generado con valores precisos que conocen los que desarrollaron la aplicación, el encargado de efectuar esta etapa, será un desarrollador.

## Verificación que las funcionalidades básicas funcionen adecuadamente.

Una vez desplegado el sistema y luego de todas las etapas anteriores, vendrá la etapa de prueba del sistema en producción, el encargado de esta etapa será el QA.

El QA procederá todas las pruebas diseñadas en los módulos con el fin de verificar si la calidad determinada en el diseño del sistema es la misma con la que se llegó a producción. Las pruebas fueron ejecutadas en ambientes de pruebas y el funcionamiento del sistema ya estaba asegurado, pero por fines de calidad final al usuario, una vez que el sistema se despliegue en el ambiente productivo, se deberán ejecutar nuevamente para capturar errores que se deban a configuraciones del ambiente.

## Elaborar plan de recuperación

Un plan de recuperación es un documento estructurado con instrucciones para responder ante incidentes no planeados. Este plan detallado paso a paso consiste en tomar las precauciones para minimizar los efectos de un desastre, de tal manera que el sistema pueda continuar operando o reanudar rápidamente las funciones operativas.

## Implementación de backup

El usuario administrador puede en cualquier momento realizar un backup de toda la base de datos del sistema, para lo cual lo primero que tendrá que hacer es iniciar sesión como administrador y luego en el sidebar hará clic en el apartado llamado backup. Dentro de ese apartado seleccionará el sub apartado llamado "ver backups", en el cual tendrá 2 botones principales: uno para efectuar un backup y otro para restaurar el backup previamente

guardado. En caso de que el usuario haga clic en el primer botón una vez ejecutado el backup se guardará el archivo correspondiente, el mismo tiene una la fecha de realización del mismo para una mejor organización y control.

Cabe destacar que el backup se comprime por la aplicación en formato .gz, esto es principalmente para ahorrar espacio, especialmente cuando se tienen cientos o miles de backups guardados.

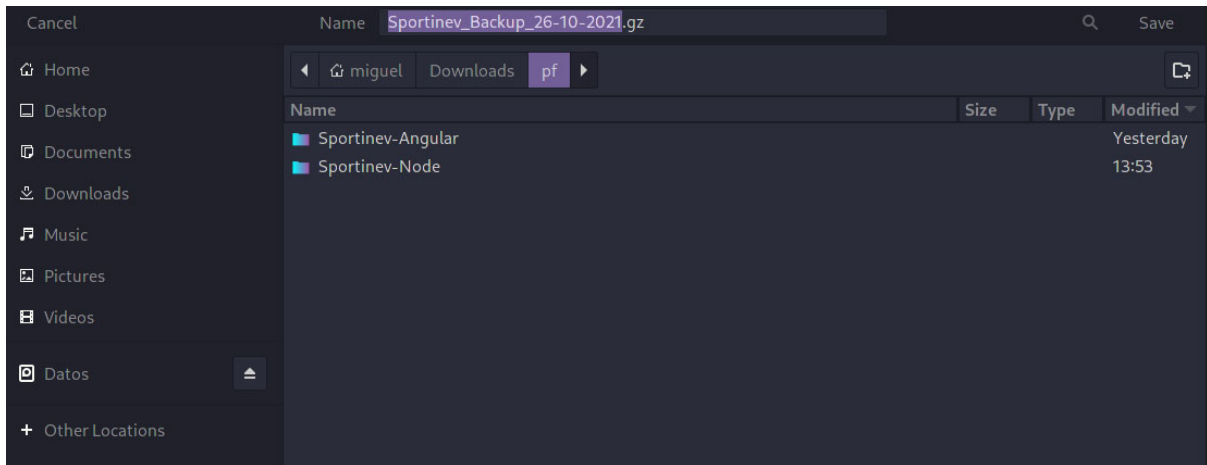


Figura 270: Funcionamiento del backup 1

## Recuperación del backup

En el caso de apretar el segundo botón el usuario seleccionará un backup previo y el sistema procederá a restaurar toda la base de datos, dicha restauración puede llevar un tiempo dependiendo de la cantidad de documentos que tenga la base de datos.

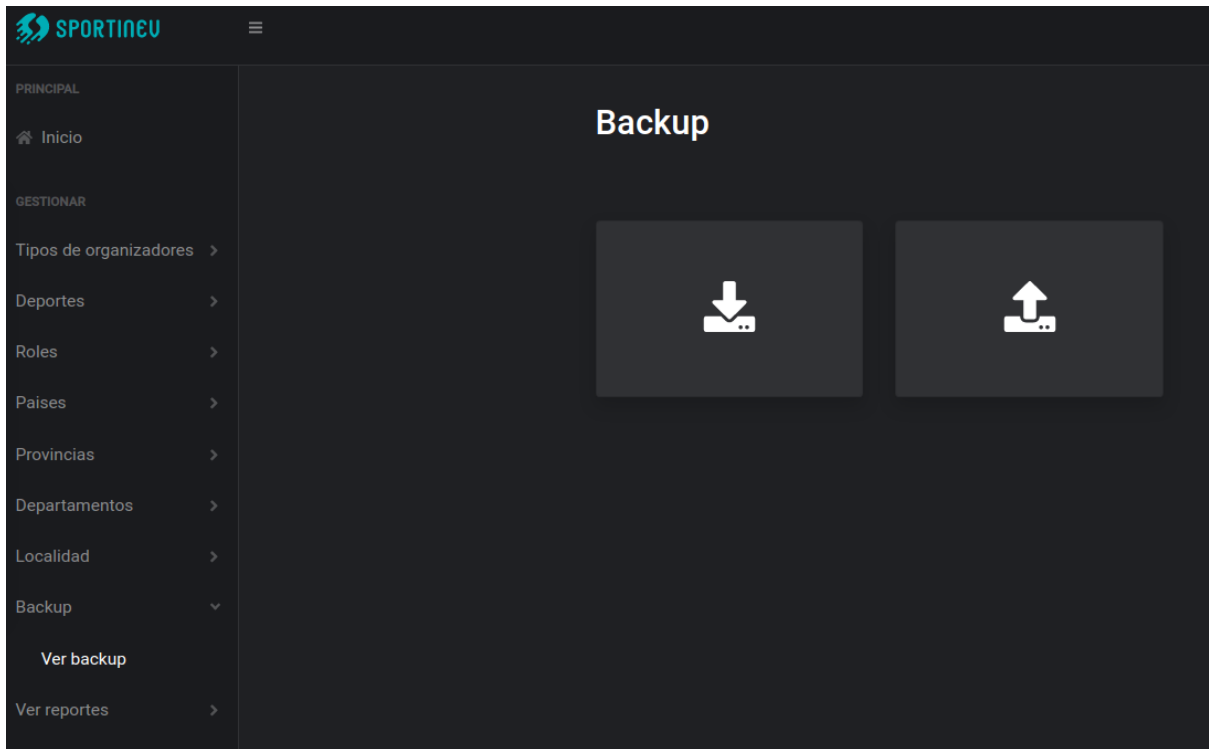


Figura 271: Funcionamiento del backup 2

## Implementación

Para la implementación del proyecto se tomó la decisión de utilizar una estructura kubernitzada, lo que nos permite utilizar en parámetros generales cualquier servicio cloud como lo son: DigitalOcean, AWS, Google Cloud, etc.

Se decidió utilizar los servicios de [Okteto](#) para realizar las pruebas internas de la aplicación y el servicio de Google Club para la puesta en producción del mismo.

### Implementación con okteto

Okteto es una plataforma de desarrollo en la nube completamente gratuita que nos permite probar la implementación del sistema en la nube.

La estructura que sigue la implementación del sistema es la siguiente:



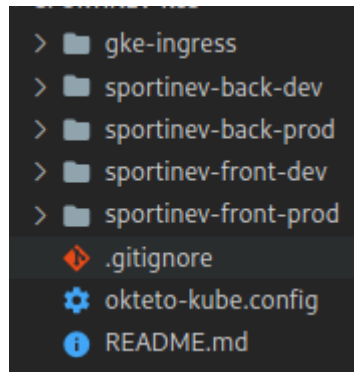


Figura 272: Imagen que muestra la estructura de la implementación

1. Archivo okteto-kube.config cuyo contenido no podemos mostrar, puesto que contiene las credenciales para conectarse al clúster.
2. Carpetas sportinev-back-dev y sportinev-front-dev que contiene la estructura sobre la cual va a ser implementado tanto el backend como el frontend del sistema en el cluster.

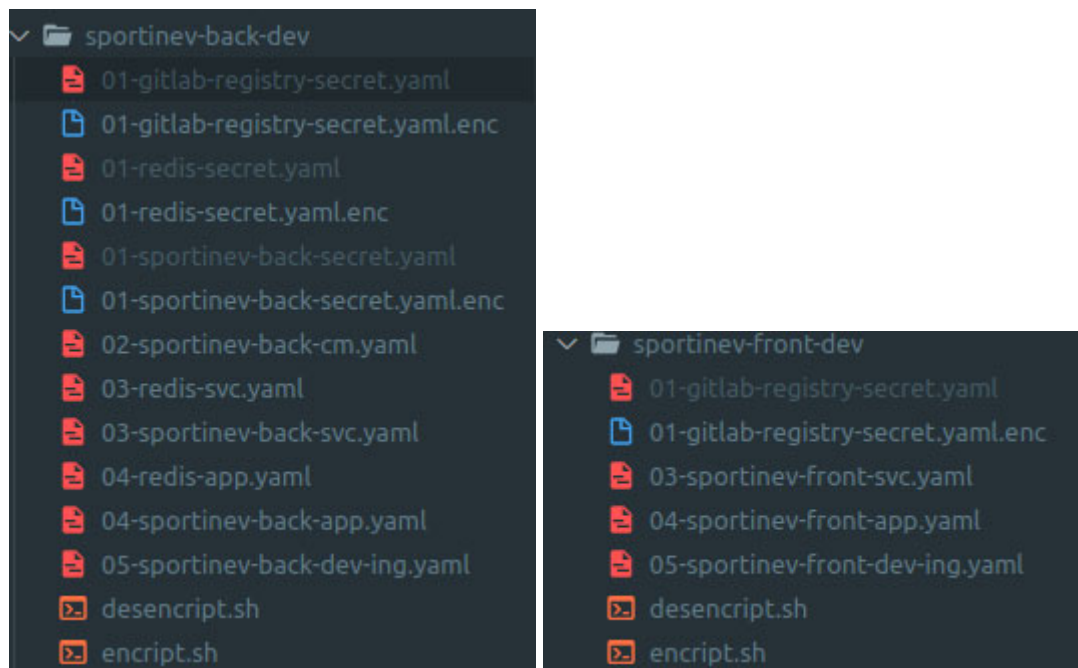


Figura 273 y 274: Imagen que muestra la estructura de la implementación de okteto

La estructura se basa en:

01-{nombre}-secret: Está destinado a guardar variables de entorno confidenciales codificadas en base64. Estas se suben a Git estando encriptadas.

# Versión de La estructura

```
apiVersion: v1
kind: ConfigMap
metadata:
  # Nombre que el configmap va a tomar dentro del cluster
  name: sportinev-back-cm
  # Espacio de nombres sobre el cual se va aplicar el configmap
  namespace: dev-sportinev
  # Variables de entorno clave: valor
data:
  PORT: "8080"
  API: https://sportinev-back-starkloff.cloud.okteto.net/
  REDIS_HOST: redis
  REDIS_PORT: "6379"
  ENVIRONMENT: prod
```

02-{nombre}-cm: Está destinado a guardar variables de entorno no confidenciales.

```
# Versión de La estructura
apiVersion: v1
kind: ConfigMap
metadata:
  # Nombre que el configmap va a tomar dentro del cluster
  name: sportinev-back-cm
  # Espacio de nombres sobre el cual se va aplicar el configmap
  namespace: dev-sportinev
  # Variables de entorno clave: valor
data:
  PORT: "8080"
  API: https://sportinev-back-starkloff.cloud.okteto.net/
  REDIS_HOST: redis
  REDIS_PORT: "6379"
  ENVIRONMENT: prod
```

03-{nombre}-svc: Una forma abstracta de exponer una aplicación que se ejecuta en un conjunto de pods como un servicio de red.

```
# Versión de La estructura
apiVersion: v1
kind: Service
metadata:
  # Nombre que el service va a tomar dentro del cluster
  name: sportinev-back
  # Espacio de nombres sobre el cual se va aplicar el service
  namespace: dev-sportinev
spec:
  # Kubernetes crea una dirección IP estable a
  # La que se puede acceder desde Los nodos del clúster.
  type: ClusterIP
  ports:
    # Protocolo sobre el cual se comunican los pods y el service
```

```
- name: http
  # Puerto del service
  port: 80
  # Variable de entorno que pasamos por el deployment, puerto del pod
  targetPort: http-server
selector:
  # Nombre que servirá para relacionar un recurso al service
  app: sportinev-back
```

04-{nombre}-app: Proporciona actualizaciones declarativas para los Pods y los ReplicaSets.

```
# Versión de La estructura
apiVersion: apps/v1
kind: Deployment
metadata:
  # Nombre que el deployment va a tomar dentro del cluster
  name: sportinev-back
  # Espacio de nombres sobre el cual se va aplicar el deployment
  namespace: dev-sportinev
  # Nivel que nos permite mapear nuestra estructura organizacional
  labels:
    # tag app
    app: sportinev-back
spec:
  # Cantidad de pods que se van a desplegar con el deployment
  replicas: 1
  selector:
    # Buscamos recurso que tenga el selector app
    matchLabels:
      # tag app
      app: sportinev-back
  # Template para conectar con variables de entorno
  template:
    metadata:
      labels:
        # tag app
        app: sportinev-back
    spec:
      containers:
        # Nombre del pod
        - name: sportinev-back
          # Imagen en donde está el proyecto
          image: registry.gitlab.com/elvio.aruta98/sportinev-node:dev
          # Cada vez que se ejecute el deployment, se consultará la imagen subida en
          # el
          # registry del repositorio
          imagePullPolicy: Always
      env:
        # Nombre de la variable de entorno
        - name: PORT
```

```
valueFrom:
  # Configmap que contiene La variable de entorno
  configMapKeyRef:
    # Nombre del recurso que contiene La variable de entorno
    name: sportinev-back-cm
    key: PORT
# Nombre de La variable de entorno
- name: API
valueFrom:
  # Configmap que contiene La variable de entorno
  configMapKeyRef:
    # Nombre del recurso que contiene La variable de entorno
    name: sportinev-back-cm
    key: API
# Nombre de La variable de entorno
- name: REDIS_HOST
valueFrom:
  # Configmap que contiene La variable de entorno
  configMapKeyRef:
    # Nombre del recurso que contiene La variable de entorno
    name: sportinev-back-cm
    key: REDIS_HOST
# Nombre de La variable de entorno
- name: REDIS_PORT
valueFrom:
  # Configmap que contiene La variable de entorno
  configMapKeyRef:
    # Nombre del recurso que contiene La variable de entorno
    name: sportinev-back-cm
    key: REDIS_PORT
# Nombre de La variable de entorno
- name: ENVIRONMENT
valueFrom:
  # Configmap que contiene La variable de entorno
  configMapKeyRef:
    # Nombre del recurso que contiene La variable de entorno
    name: sportinev-back-cm
    key: ENVIRONMENT
# Nombre de La variable de entorno
- name: MONGO_URL
valueFrom:
  # Secret que contiene La variable de entorno
  secretKeyRef:
    # Nombre del recurso que contiene La variable de entorno
    name: sportinev-back-secret
    key: MONGO_URL
# Nombre de La variable de entorno
- name: EMAIL
valueFrom:
  # Secret que contiene La variable de entorno
  secretKeyRef:
    # Nombre del recurso que contiene La variable de entorno
    name: sportinev-back-secret
```

```

    key: EMAIL
  # Nombre de la variable de entorno
- name: PASSWORD_MAIL
  valueFrom:
    # Secret que contiene la variable de entorno
    secretKeyRef:
      # Nombre del recurso que contiene la variable de entorno
      name: sportinev-back-secret
      key: PASSWORD_MAIL
  # Nombre de la variable de entorno
- name: REDIS_PASSWORD
  valueFrom:
    # Secret que contiene la variable de entorno
    secretKeyRef:
      # Nombre del recurso que contiene la variable de entorno
      name: sportinev-back-secret
      key: REDIS_PASSWORD
  # Nombre de la variable de entorno
- name: SECRETOPRIVATEKEY
  valueFrom:
    # Secret que contiene la variable de entorno
    secretKeyRef:
      # Nombre del recurso que contiene la variable de entorno
      name: sportinev-back-secret
      key: SECRETOPRIVATEKEY
  # Nombre de la variable de entorno
- name: MERCADOPAGOTOKEN
  valueFrom:
    # Secret que contiene la variable de entorno
    secretKeyRef:
      # Nombre del recurso que contiene la variable de entorno
      name: sportinev-back-secret
      key: MERCADOPAGOTOKEN
ports:
  # Variable de entorno que pasamos por el puerto del pod
- name: http-server
  # Puerto del pod donde está montado el backend
  containerPort: 8080
# Comando para extraer las imágenes del registry de gitlab y no de dockerhub
imagePullSecrets:
  # Secret que contiene las credenciales para acceder al registry de
  gitlab
- name: gitlab-registry

```

05-{nombre}-ing: Administra el acceso externo a los servicios en un clúster y se encarga de aplicar el certificado SSL para exponer como https.

```

# Versión de la estructura
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:

```

```

# Indicamos a Okteto que genere la ruta de acceso
annotations:
  dev.okteto.com/generate-host: "true"
# Nombre que el service va a tomar dentro del cluster
name: sportinev-back
# Espacio de nombres sobre el cual se va a aplicar el service
namespace: dev-sportinev
spec:
  rules:
    - http:
      paths:
        # Combinación de nombres de puerto y servicio
        - backend:
            # Nombre del servicio que apunta a los pods generados por el deployment
            serviceName: sportinev-back
            # Puerto del dominio donde se expondrán los pods
            servicePort: 80
  
```

Como resultado obtenemos:

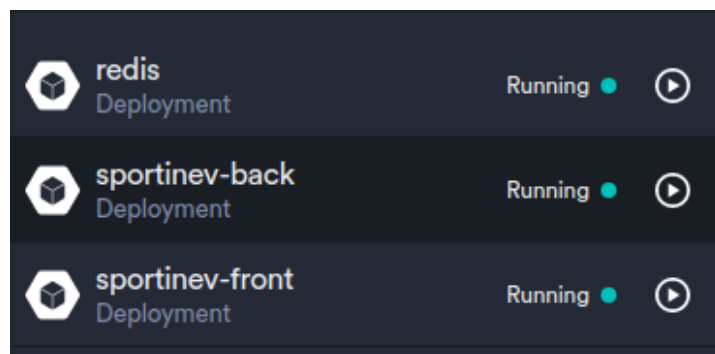


Figura 275: Imagen que muestra el resultado obtenido en okteto

Para conectarnos con el clúster debemos

- Tener instalado
  - [Kubect!](#)

Configurar la variable de entorno KUBECONFIG

```
$ export KUBECONFIG=${HOME}/{path_to_okteto_kube_config}:${KUBECONFIG:-$HOME/.kube/config}
```

Por último tenemos implementado en GitLab el CI/CD que nos permite crear, probar, implementar y monitorear automáticamente nuestras aplicaciones.

Para ello primero debemos configurar las variables de entorno en GitLab para poder desplegar automáticamente los cambios realizados sobre la imagen en el cluster. Debemos dirigirnos al apartado settings del proyecto y seleccionar la opción CI/CD

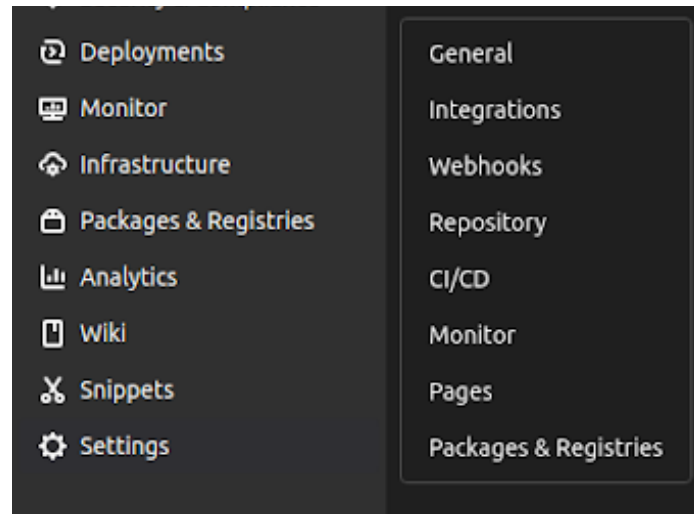


Figura 276: Imagen que muestra la estructura de la implementación

Allí debemos expandir el campo Variables y agregar las siguientes variables de entorno

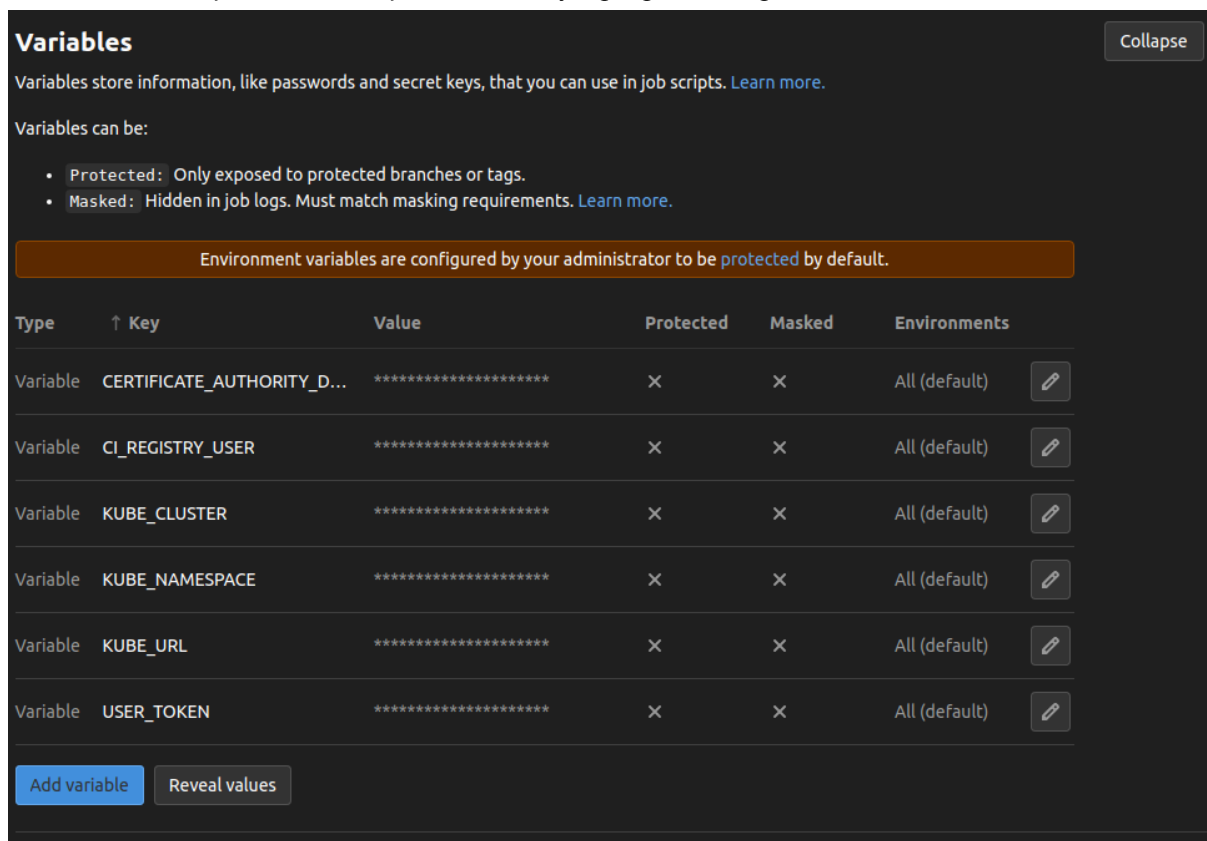


Figura 277: Imagen que muestra la configuración de las variables

Ahora procedemos a configurar el CI/CD para que cada vez que se hagan cambios sobre la rama master se actualice el deployment con la nueva versión del sistema.

```
# Utilizamos docker como imagen por defecto
image: docker:latest
services:
  - docker:dind

# Etapas del trabajo
stages:
  # Build de la aplicación y push de la imagen al registry
  - package
  # Actualizar el deployment con la imagen actualizada
  - deploy

docker-build:
  stage: package
  script:
    # Construimos la imagen con tag dev
    - docker build -t registry.gitlab.com/elvio.aruta98/sportinev-node:dev .
    # Vinculamos docker con gitlab para poder pushear la imagen al registry del
    repositorio
    - docker login -u gitlab-ci-token -p $CI_BUILD_TOKEN registry.gitlab.com
    # Pusheamos la nueva imagen al registry
    - docker push registry.gitlab.com/elvio.aruta98/sportinev-node:dev
  only:
    # Solo se ejecuta en master
    - master

k8s-deploy:
  image: dtzar/helm-kubect1
  stage: deploy
  script:
    # Construimos el certificado que nos identifica como usuario autorizado en el cluster
    - echo "$CERTIFICATE_AUTHORITY_DATA" > "$(pwd)/cloud_okteto_com.pem"
    # Configuramos conexión con el cluster
    - kubectl config set-cluster "$KUBE_CLUSTER" --server="$KUBE_URL"
    --certificate-authority="$(pwd)/cloud_okteto_com.pem"
    # Configuramos las credenciales del usuario
    - kubectl config set-credentials "$CI_REGISTRY_USER" --token="$USER_TOKEN"
    # Generamos el contexto
    - kubectl config set-context "$KUBE_NAMESPACE"@"$KUBE_CLUSTER"
    --cluster="$KUBE_CLUSTER" --namespace="$KUBE_NAMESPACE" --user="$CI_REGISTRY_USER"
    # Usamos el contexto que creamos
    - kubectl config use-context "$KUBE_NAMESPACE"@"$KUBE_CLUSTER"
    # Imprimimos la configuración para ver si es acertada
    - kubectl config view
    # Reiniciamos el deployment
    - kubectl rollout restart deployment.apps/sportinev-back
  only:
    # Solo se ejecuta en master
    - master
```



### Implementación con Google Cloud Platform

Google Cloud es una plataforma virtual desarrollada por Google a través del cual se puede realizar una serie de tareas que antes requerían de hardware o software y que ahora utilizan la nube de Google como única forma de acceso, almacenamiento y gestión de datos.

La estructura que sigue la implementación del sistema es la siguiente:

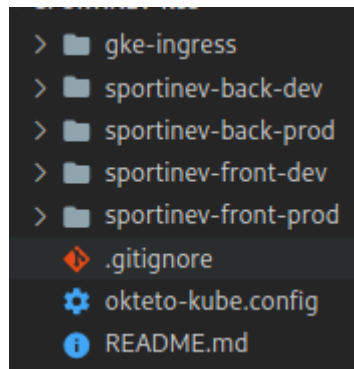


Figura 278: Imagen que muestra la estructura de la implementación

3. Archivo okteto-kube.config cuyo contenido no podemos mostrar, puesto que contiene las credenciales para conectarse al clúster.
4. Carpetas sportinev-back-dev y sportinev-front-dev que contiene la estructura sobre la cual va a ser implementado, tanto el backend, como el frontend, del sistema en el cluster.

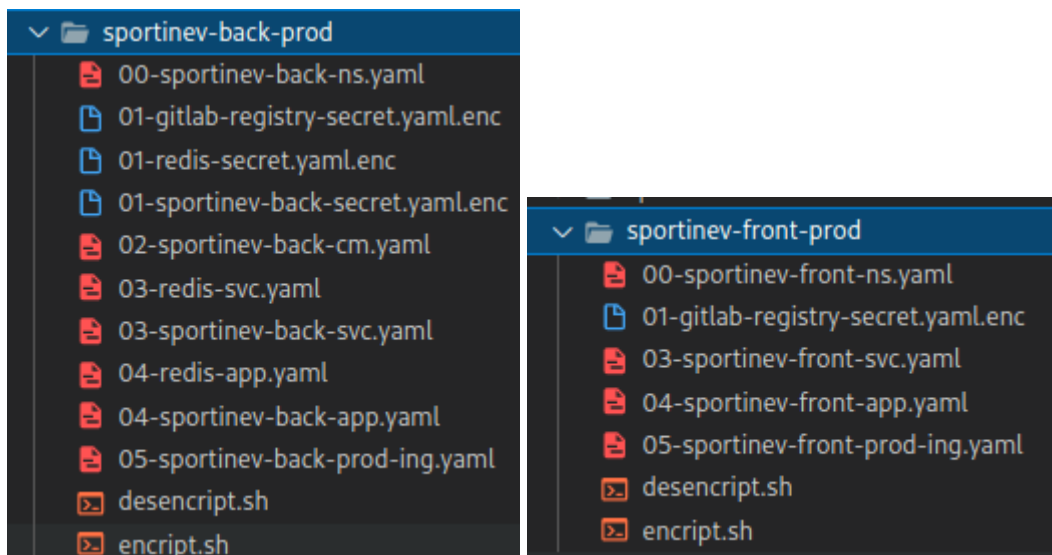


Figura 279 y 280: Imagen que muestra la estructura de la implementación de google cloud platform

La estructura de google cloud platform está basada en:

01-{nombre}-secret: Está destinado a guardar variables de entorno confidenciales codificadas en base64. Estas se suben a git estando encriptadas.

```
# Versión de La estructura
apiVersion: v1
kind: Secret
metadata:
  # Nombre que el secret va a tomar dentro del cluster
  name: sportinev-back-secret
  # Espacio de nombres sobre el cual se va aplicar el secret
  namespace: dev-sportinev
# Datos arbitrarios definidos por el usuario
type: Opaque
# Variables de entorno clave: valor
data:
  MONGO_URL:
    bW9uZ29kYitzcnY6Ly9zdGFya2xvZmY6d00zbV10UmJBYk9wVkJFY0BjbHVzdGVyc3BvcnRpbmV2LjQ5dGZpLm1v
    bmdvZGIubmV0L3Nwb3J0aW5ldkRC
  EMAIL: c3BvcnRpbmV2QGdtYWlsLmNvbQ==
  PASSWORD_MAIL: eWZibWpnc2RiYmRncXR3dQ==
  REDIS_PASSWORD: SUpjTGFESGhJU9RSzY0K0VGbkGxY1NCT1hvTGxZL1lGNHByOE41ZFExWQ==
  SECRETOPRIVATEKEY: Y0c/PWUtOF5NfjIk
  MERCADOPAGOTOKEN:
    VEVTVC04MzcyNjMxNzYxNzkzNDg3LTA4MTkyMS02MTg1YTJiYjQzMjg4OWE3Nm1YzUzMzRkMDAxMGRkMi0zNTY0
    ODQxNzQ=
```

02-{nombre}-cm: Está destinado a guardar variables de entorno no confidenciales.

```
# Versión de La estructura
apiVersion: v1
kind: ConfigMap
metadata:
  # Nombre que el configmap va a tomar dentro del cluster
  name: sportinev-back-cm
  # Espacio de nombres sobre el cual se va aplicar el configmap
  namespace: sportinev-back-prod
# Variables de entorno clave: valor
data:
  PORT: "8080"
  API: https://backend.sportinev.com/
  REDIS_HOST: redis
  REDIS_PORT: "6379"
  ENVIRONMENT: prod
```

03-{nombre}-svc: Una forma abstracta de exponer una aplicación que se ejecuta en un conjunto de pods como un servicio de red.

```
apiVersion: v1
```

```
kind: Service
metadata:
  name: redis
  namespace: sportinev-back-prod
  labels:
    app: redis
spec:
  type: ClusterIP
  selector:
    app: redis
  ports:
  - name: http
    port: 6379
    targetPort: http-server
```

```
apiVersion: v1
kind: Service
metadata:
  name: sportinev-front
  namespace: sportinev-front-prod
spec:
  type: ClusterIP
  ports:
  - name: http
    port: 80
    targetPort: http-server
  selector:
    app: sportinev-front
```

04-{nombre}-app: Proporciona actualizaciones declarativas para los Pods y los ReplicaSets.

```
# Versión de la estructura
apiVersion: apps/v1
kind: Deployment
metadata:
  # Nombre que el deployment va a tomar dentro del cluster
  name: sportinev-back
  # Espacio de nombres sobre el cual se va a aplicar el deployment
  namespace: sportinev-back-prod
  # Nivel que nos permite mapear nuestra estructura organizacional
  labels:
    # tag app
    app: sportinev-back
spec:
  # Cantidad de pods que se van a desplegar con el deployment
  replicas: 1
  selector:
```

```
# Buscamos recurso que tenga el selector app
matchLabels:
  # tag app
  app: sportinev-back
# Template para conectar con variables de entorno
template:
  metadata:
    labels:
      # tag app
      app: sportinev-back
  spec:
    containers:
      # Nombre del pod
      - name: sportinev-back
        # Imagen en donde esta el proyecto
        image: registry.gitlab.com/elvio.aruta98/sportinev-node:prod
        # Cada vez que se ejecute el deployment, se consultara La imagen subida en el
        registry del repositorio
        imagePullPolicy: Always
      env:
        # Nombre de La variable de entorno
        - name: PORT
          valueFrom:
            # Configmap que contiene La variable de entorno
            configMapKeyRef:
              # Nombre del recurso que contiene La variable de entorno
              name: sportinev-back-cm
              key: PORT
        # Nombre de La variable de entorno
        - name: API
          valueFrom:
            # Configmap que contiene La variable de entorno
            configMapKeyRef:
              # Nombre del recurso que contiene La variable de entorno
              name: sportinev-back-cm
              key: API
        # Nombre de La variable de entorno
        - name: REDIS_HOST
          valueFrom:
            # Configmap que contiene La variable de entorno
            configMapKeyRef:
              # Nombre del recurso que contiene La variable de entorno
              name: sportinev-back-cm
              key: REDIS_HOST
        # Nombre de La variable de entorno
        - name: REDIS_PORT
          valueFrom:
            # Configmap que contiene La variable de entorno
            configMapKeyRef:
              # Nombre del recurso que contiene La variable de entorno
              name: sportinev-back-cm
              key: REDIS_PORT
        # Nombre de La variable de entorno
```

```
- name: ENVIRONMENT
valueFrom:
  # Configmap que contiene La variable de entorno
  configMapKeyRef:
    # Nombre del recurso que contiene La variable de entorno
    name: sportinev-back-cm
    key: ENVIRONMENT
# Nombre de La variable de entorno
- name: MONGO_URL
valueFrom:
  # Secret que contiene La variable de entorno
  secretKeyRef:
    # Nombre del recurso que contiene La variable de entorno
    name: sportinev-back-secret
    key: MONGO_URL
# Nombre de La variable de entorno
- name: EMAIL
valueFrom:
  # Secret que contiene La variable de entorno
  secretKeyRef:
    # Nombre del recurso que contiene La variable de entorno
    name: sportinev-back-secret
    key: EMAIL
# Nombre de La variable de entorno
- name: PASSWORD_MAIL
valueFrom:
  # Secret que contiene La variable de entorno
  secretKeyRef:
    # Nombre del recurso que contiene La variable de entorno
    name: sportinev-back-secret
    key: PASSWORD_MAIL
# Nombre de La variable de entorno
- name: REDIS_PASSWORD
valueFrom:
  # Secret que contiene La variable de entorno
  secretKeyRef:
    # Nombre del recurso que contiene La variable de entorno
    name: sportinev-back-secret
    key: REDIS_PASSWORD
# Nombre de La variable de entorno
- name: SECRETPRIVATEKEY
valueFrom:
  # Secret que contiene La variable de entorno
  secretKeyRef:
    # Nombre del recurso que contiene La variable de entorno
    name: sportinev-back-secret
    key: SECRETPRIVATEKEY
# Nombre de La variable de entorno
- name: MERCADOPAGOTOKEN
valueFrom:
  # Secret que contiene La variable de entorno
  secretKeyRef:
    # Nombre del recurso que contiene La variable de entorno
```

```

        name: sportinev-back-secret
        key: MERCADOPAGOTOKEN
# Nombre de La variable de entorno
- name: CLOUDINARY_NAME
  valueFrom:
    # Secret que contiene La variable de entorno
    secretKeyRef:
      # Nombre del recurso que contiene La variable de entorno
      name: sportinev-back-secret
      key: CLOUDINARY_NAME
# Nombre de La variable de entorno
- name: CLOUDINARY_API_KEY
  valueFrom:
    # Secret que contiene La variable de entorno
    secretKeyRef:
      # Nombre del recurso que contiene La variable de entorno
      name: sportinev-back-secret
      key: CLOUDINARY_API_KEY
# Nombre de La variable de entorno
- name: CLOUDINARY_API_SECRET
  valueFrom:
    # Secret que contiene La variable de entorno
    secretKeyRef:
      # Nombre del recurso que contiene La variable de entorno
      name: sportinev-back-secret
      key: CLOUDINARY_API_SECRET
ports:
  # Variable de entorno que pasamos por el puerto del pod
  - name: http-server
    # Puerto del pod donde esta montado el backend
    containerPort: 8080
# Comando para extraer Las imágenes del registry de gitlab y no de dockerhub
imagePullSecrets:
  # Secret que contiene Las credenciales para acceder al registry de gitlab
  - name: gitlab-registry

```

05-{nombre}-ing: Administra el acceso externo a los servicios en un clúster y se encarga de aplicar el certificado SSL para exponer como https.

```

# Versión de La estructura
apiVersion: networking.k8s.io/v1beta1
kind: Ingress
metadata:
  # Utilizamos certificados que estan alojados en Letsencrypt-prod
  annotations:
    kubernetes.io/ingress.class: nginx
    cert-manager.io/cluster-issuer: "letsencrypt-prod"
    nginx.ingress.kubernetes.io/ssl-redirect: 'true'
    nginx.ingress.kubernetes.io/rewrite-target: /$1

```

```

# Nombre que el service va a tomar dentro del cluster
name: sportinev-back
# Espacio de nombres sobre el cual se va aplicar el service
namespace: sportinev-back-prod
spec:
  tls:
    - hosts:
      - backend.sportinev.com
      secretName: sportinev-back-tls
  rules:
    - host: backend.sportinev.com
      http:
        paths:
          # Combinación de nombres de puerto y servicio
          - backend:
              # Nombre del servicio que apunta a los pods generados por el deployment
              serviceName: sportinev-back
              # Puerto del dominio donde se expondrán los pods
              servicePort: 80
              path: /(.*)
  
```

Como resultado obtenemos:

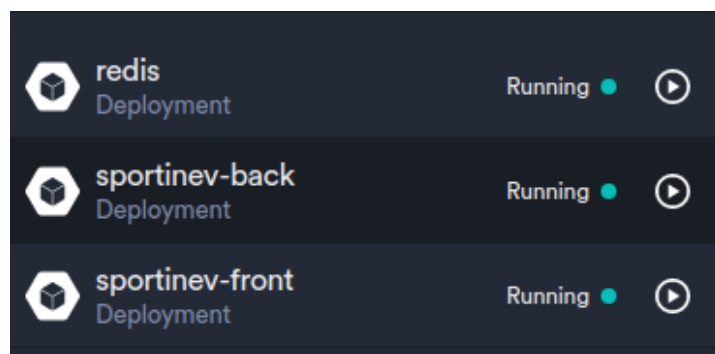


Figura 281: Imagen que muestra la el resultado obtenido en okteto

Para conectarnos con el cluster debemos al igual que en okteto tendremos que tener instalado kubectl y configurar la variable de entorno KUBECONFIG.

Un gran diferencia entre okteto y google cloud platform es que con gcp tenemos namespaces personalizables, mientras que en okteto esto es imposible, ya que solamente nos dan uno, es por esto que necesitamos un archivo .yaml mostrado a continuación para configurar los namespaces

```

apiVersion: v1
kind: Namespace
metadata:
  creationTimestamp: null
  
```

```
name: sportinev-back-prod
spec: {}
status: {}
```

```
apiVersion: v1
kind: Namespace
metadata:
  creationTimestamp: null
  name: sportinev-front-prod
spec: {}
status: {}
```

Otra diferencia entre okteto y google cloud platform es con el componente ingress, el cual estamos utilizando los certificados de let's encrypt en el cual le especificamos el dominio que compramos llamado sportinev.com.

```
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
  name: letsencrypt-prod
spec:
  acme:
    # The ACME server URL
    server: https://acme-v02.api.letsencrypt.org/directory
    # Email address used for ACME registration
    email: sportinev@gmail.com
    # Name of a secret used to store the ACME account private key
    privateKeySecretRef:
      name: letsencrypt-prod
    # Enable the HTTP-01 challenge provider
    solvers:
      - http01:
          ingress:
            class: nginx
```

## Puestos de trabajo

Cada uno de los integrantes del equipo trabajó en forma remota con sus computadores personales, obviamente cada una de estas computadoras cumplía con los requisitos mínimos especificados anteriormente. También se contó un lugar cómodo y tranquilo para aumentar la productividad de cada uno de los integrantes.



## Métodos de replicación y replicación de implementación en otros sitios

El sistema es muy fácil de replicar en distintos servicios o dominios, ya que al estar utilizando Kubernetes. El paso de un cloud provider a otro es mucho más ameno, ya que es un formato común que muchos servicios cloud emplean, obviamente que a medida que se van complicando, vamos a empezar a tener un vendor lock-in.

En cuanto a los métodos de replicación de la aplicación, en caso de que ocurra algún problema con la implementación de algunos de los servicios podemos usar el otro sin ningún problema. Por ej. si por algún motivo el servicio de google cloud platform deja de funcionar se podría redirigir temporalmente a los usuarios al dominio de okteto, así de esta forma el usuario final puede tener un acceso 24/7 al sistema. Obviamente que en este caso se podrá considerar comprar una suscripción a una cuenta prémium de okteto para soportar el tráfico de la aplicación mientras la plataforma de Google se restablece.

## Capacitación del administrador

Una vez el sistema esté desplegado y funcionando correctamente, se capacitará a los usuarios siguiendo el plan de capacitación definido anteriormente.

En esta etapa de implementación se hará hincapié en la capacitación del administrador, explicando las principales funcionalidades que tiene el administrador en el sistema.












Como mencionado en el plan de capacitación las capacitaciones que se le darán al administrador del sistema son las siguientes:




















- Manejo de la sección Perfil, cerrar sesión, darse de baja.
- Manejo de: Tipos de organizadores, estado inscripción evento, estado evento, estado patrocinador del evento, tipos de deportes, tipos de eventos.
- Alta, baja y modificación de deportes.
- Dar de alta un rol con sus permisos.
- Asignar roles de Administrador a otro usuario.


















## Bibliografía



1. Eventols (27 de abril de 2021). Recuperado de: <https://eventols.com/>
2. Sporteasy (27 de abril de 2021). Recuperado de: <https://www.sporteasy.net/es/home/>
3. EventBase (27 de abril de 2021). Recuperado de: <https://www.eventbase.com/>
4. Playinga (27 de abril de 2021). Recuperado de: <https://playinga.com/es/>
5. Playinga android (27 de abril de 2021). Recuperado de: <https://apps.apple.com/us/app/thapos/id1103693671>
6. Playinga apple (27 de abril de 2021). Recuperado de: <https://apps.apple.com/us/app/thapos/id1103693671>
7. Sporteasy android (27 de abril de 2021). Recuperado de: <https://play.google.com/store/apps/details?id=com.sporteasy.android>
8. Sporteasy apple (27 de abril de 2021). Recuperado de: <https://apps.apple.com/fr/app/sporteasy-organisez-vos-matches/id599931266?l=en>
9. Angular (27 de abril de 2021). Recuperado de: <https://angular.io/docs>
10. Node.js (27 de abril de 2021). Recuperado de: <https://nodejs.org/en/docs/>
11. Express.js (27 de abril de 2021). Recuperado de: <https://expressjs.com/>
12. Kubernetes (27 de abril de 2021). Recuperado de: <https://kubernetes.io/docs/home/>
13. Docker (27 de abril de 2021). Recuperado de: <https://docs.docker.com/>
14. MongoDB (27 de abril de 2021). Recuperado de: <https://www.mongodb.com/>
15. WhatsApp (27 de abril de 2021). Recuperado de: <https://www.whatsapp.com/features>
16. Discord (27 de abril de 2021). Recuperado de: <https://discord.com/>
17. Google Drive (27 de abril de 2021). Recuperado de: <https://www.blog.google/products/drive/>
18. Moodle (27 de abril de 2021). Recuperado de: [https://docs.moodle.org/all/es/Acerca\\_de\\_Moodle](https://docs.moodle.org/all/es/Acerca_de_Moodle)
19. Trello (27 de abril de 2021). Recuperado de: <https://blog.trello.com/es>
20. GIT (27 de abril de 2021). Recuperado de: <https://git-scm.com/doc>
21. Visual paradigm (27 de abril de 2021). <https://online.visual-paradigm.com/>

		Name	Start	Duration	Finish	Predecessors	Resource Names
1		<b>Definicion de Empresa y Sistema</b>	<b>3/1/21, 8:00 AM</b>	<b>15 days</b>	<b>3/19/21, 5:00 PM</b>		
2		Definir sistema	3/1/21, 8:00 AM	15 days	3/19/21, 5:00 PM		Scrum Master;Product...
3		ENTREGABLES:Documentación "TP 1" en carpeta Proyecto Final / TP 1 en Google Dr...	3/1/21, 8:00 AM	0 days	3/1/21, 8:00 AM		
4		<b>Relevamiento General</b>	<b>3/19/21, 5:00 PM</b>	<b>6 days</b>	<b>3/29/21, 5:00 PM</b>	<b>1</b>	
5		Realizar estudio sobre el rubro	3/22/21, 8:00 AM	3 days	3/24/21, 5:00 PM		Product owner;Scrum ...
6		Realizar relevamiento sobre las organizaciones del rubro	3/25/21, 8:00 AM	3 days	3/29/21, 5:00 PM	5	Analista 1
7		Detectar funciones que realizan los sistemas de la competencia	3/25/21, 8:00 AM	3 days	3/29/21, 5:00 PM	5	Analista 2
8		Realizar un relevamiento sobre las tecnologías que usa la competencia	3/25/21, 8:00 AM	3 days	3/29/21, 5:00 PM	5	Analista 3
9		ENTREGABLES:Documentación "TP 1" en carpeta Proyecto Final / TP 1 en Google Dr...	3/19/21, 5:00 PM	0 days	3/19/21, 5:00 PM		
10		<b>Relevamiento detallado y análisis del Sistema.</b>	<b>3/29/21, 5:00 PM</b>	<b>14 days?</b>	<b>4/16/21, 5:00 PM</b>	<b>4</b>	
11		Detallar y documentar de forma detallada todas la funciones antes relevadas	3/30/21, 8:00 AM	7 days	4/7/21, 5:00 PM		Analista 1
12		Realizar modelo logico de sistemas de competencia	3/30/21, 8:00 AM	7 days	4/7/21, 5:00 PM		Analista 2
13		Detallar las necesidades no cubiertas por la competencia	3/30/21, 8:00 AM	6 days	4/6/21, 5:00 PM		Analista 3
14		Realizar reunion con partes interesadas	4/7/21, 8:00 AM	1 day?	4/7/21, 5:00 PM	13	Product owner
15		Detallar los objetivos y alcances preliminares del nuevo Sistema	4/8/21, 8:00 AM	7 days	4/16/21, 5:00 PM	11;12;13	Scrum Master
16		ENTREGABLES:Documentación "TP 1" en carpeta Proyecto Final / TP 1 en Google Dr...	3/29/21, 5:00 PM	0 days	3/29/21, 5:00 PM		
17		<b>Planificación</b>	<b>4/16/21, 5:00 PM</b>	<b>16 days?</b>	<b>5/10/21, 5:00 PM</b>	<b>10</b>	
18		Estimar tiempos y recursos de las actividades	4/19/21, 8:00 AM	2 days	4/20/21, 5:00 PM		Scrum Master
19		Seleccionar metodologias a utilizar	4/19/21, 8:00 AM	2 days	4/20/21, 5:00 PM		Analista 1
20		Definir roles de los integrantes del equipo	4/21/21, 8:00 AM	2 days	4/22/21, 5:00 PM		Scrum Master
21		Definir las tecnologías a utilizar	4/19/21, 8:00 AM	2 days	4/20/21, 5:00 PM		Analista 2;Analista 3

		Name	Start	Duration	Finish	Predecessors	Resource Names
22		Definir herramientas de comunicación	4/21/21, 8:00 AM	0.5 days	4/21/21, 1:00 PM		Analista 2
23		Definir herramientas para el seguimiento de tareas	4/21/21, 3:00 PM	0.5 days	4/22/21, 10:00 AM		Analista 3
24		Definir herramientas para la gestión de configuración del software	4/21/21, 8:00 AM	0.5 days	4/21/21, 1:00 PM		Analista 1
25		Definir recursos para cada una de las actividades	4/23/21, 8:00 AM	1 day	4/23/21, 5:00 PM	18;20	Scrum Master
26		Realizar reunion con partes interesadas	4/24/21, 8:00 AM	1 day?	4/26/21, 5:00 PM		Product owner
27		Realizar estudio de factibilidad	5/5/21, 8:00 AM	2 days	5/6/21, 5:00 PM		Scrum Master;Product...
28		Estimar costos de infraestructura y tecnologías	5/5/21, 8:00 AM	1.5 days	5/6/21, 1:00 PM		Analista 2
29		Realizar análisis de riesgos	5/7/21, 8:00 AM	2 days	5/10/21, 5:00 PM		Analista 3
30		Realizar análisis de impacto ambiental	5/5/21, 8:00 AM	2 days	5/6/21, 5:00 PM		Scrum Master
31		ENTREGABLES: Diagrama de Tiempos, Documentación "TP 2" en carpeta Proyecto ...	4/16/21, 5:00 PM	0 days	4/16/21, 5:00 PM		
32		<b>Diseño</b>	<b>5/10/21, 5:00 PM</b>	<b>21 days?</b>	<b>6/8/21, 5:00 PM</b>	<b>17</b>	
33		Detallar los objetivos y alcances definitivos del nuevo Sistema.	5/11/21, 8:00 AM	2 days	5/12/21, 5:00 PM		Scrum Master
34		Definir salidas del sistema	5/11/21, 8:00 AM	2 days	5/12/21, 5:00 PM		Diseñador 1
35		Realizar mockups del sistema	5/11/21, 8:00 AM	2 days	5/12/21, 5:00 PM		Diseñador 2
36		Realizar el modelado de los datos	5/11/21, 8:00 AM	2 days	5/12/21, 5:00 PM		Diseñador 3
37		Definir los casos de uso del sistema	5/11/21, 8:00 AM	2 days	5/12/21, 5:00 PM		Product owner;Scrum ...
38		Definir las historias de usuario	5/11/21, 8:00 AM	2 days	5/12/21, 5:00 PM		Analista 1
39		Estimar la duración de las historias de usuario	5/13/21, 8:00 AM	1 day	5/13/21, 5:00 PM	38	Diseñador 1;Diseñador 2
40		Definir ceremonias que se realizaran como parte de las metodologías ágiles	5/11/21, 8:00 AM	0.5 days	5/11/21, 1:00 PM		Diseñador 2
41		Realizar reunion con partes interesadas	5/11/21, 8:00 AM	1 day?	5/11/21, 5:00 PM		Product owner
42		Realizar diseño del modulo de sharing	5/11/21, 8:00 AM	1.5 days?	5/12/21, 1:00 PM		Diseñador 1

		Name	Start	Duration	Finish	Predecessors	Resource Names
43		Realizar diseño del modulo de seguridad	5/12/21, 8:00 AM	1.5 days	5/13/21, 1:00 PM		Diseñador 1;Diseñador...
44		Realizar diseño del modulo de eventos	5/14/21, 8:00 AM	1.5 days	5/17/21, 1:00 PM		Diseñador 1;Diseñador...
45		Realizar diseño del modulo de reportes	5/18/21, 8:00 AM	1.5 days	5/19/21, 1:00 PM		Diseñador 1;Diseñador...
46		Realizar diseño del modulo de publicidad	5/20/21, 8:00 AM	1.5 days	5/21/21, 1:00 PM		Diseñador 1;Diseñador...
47		Realizar diseño del modulo de inscripcion	5/22/21, 8:00 AM	1.5 days	5/25/21, 1:00 PM		Diseñador 1;Diseñador...
48		Realizar diseño del modulo de pagos	5/26/21, 8:00 AM	1.5 days	5/27/21, 1:00 PM		Diseñador 1;Diseñador...
49		Realizar diseño del modulo de feedback	5/28/21, 8:00 AM	1.5 days	5/31/21, 1:00 PM		Diseñador 1;Diseñador...
50		Realizar diseño del modulo de parametrizacion	6/1/21, 8:00 AM	1.5 days	6/2/21, 1:00 PM		Diseñador 1;Diseñador...
51		Realizar diseño del modulo de perfil	6/2/21, 8:00 AM	1.5 days?	6/3/21, 1:00 PM		Scrum Master;Product...
52		Realizar diseño del modulo de recomendación	6/3/21, 8:00 AM	1.5 days	6/4/21, 1:00 PM		Diseñador 1;Diseñador...
53		Diseñar las pruebas de unidad	5/26/21, 8:00 AM	2 days?	5/27/21, 5:00 PM		QA
54		Diseñar las pruebas de integración	5/28/21, 8:00 AM	2 days	5/31/21, 5:00 PM	53	QA
55		Realizar pruebas de seguridad por niveles de usuario	6/3/21, 8:00 AM	2 days	6/4/21, 5:00 PM	54	QA
56		Diseñar pruebas de carga	6/7/21, 8:00 AM	2 days	6/8/21, 5:00 PM	55	QA
57		ENTREGABLES: Diagrama de Clases, Diagrama de Casos de Uso, Documentación "T...	5/10/21, 5:00 PM	0 days?	5/10/21, 5:00 PM		
58		<b>Arreglos de diseño</b>	<b>6/9/21, 8:00 AM</b>	<b>43 days?</b>	<b>8/6/21, 5:00 PM</b>	<b>32</b>	
59		Realizar capacitaciones	6/9/21, 8:00 AM	20 days	7/6/21, 5:00 PM		Desarrollador Back-En...
60		Realizar actividades para fomentar la comunicacion	7/7/21, 8:00 AM	1 day	7/7/21, 5:00 PM		Desarrollador Back-En...
61		Realizar actividades motivacionales	7/8/21, 8:00 AM	1 day	7/8/21, 5:00 PM		Desarrollador Back-En...
62		Analisis de proyectos anteriores	7/9/21, 8:00 AM	2 days	7/12/21, 5:00 PM		Scrum Master;Product...
63		Realizar reunion con partes interesadas	7/13/21, 8:00 AM	1 day?	7/13/21, 5:00 PM		Product owner

		Name	Start	Duration	Finish	Predecessors	Resource Names
64		Realizar revisiones de diseño colaborativo	7/14/21, 8:00 AM	8 days	7/23/21, 5:00 PM	62	Analista 1;Diseñador 1...
65		Realizar correcciones de diseño	7/26/21, 8:00 AM	10 days	8/6/21, 5:00 PM	64	Analista 1;Diseñador 1...
66		<b>Desarrollo e implementación</b>	<b>8/9/21, 8:00 AM</b>	<b>61.25 days?</b>	<b>11/2/21, 10:00 AM</b>	<b>65</b>	
67		Planificar implementación	8/9/21, 8:00 AM	1 day	8/9/21, 5:00 PM		Scrum Master;Desarro...
68		Realizar reunion con partes interesadas	8/9/21, 8:00 AM	1 day	8/9/21, 5:00 PM		Product owner
69		Realizar analisis economico de la arquitectura	8/9/21, 8:00 AM	1 day	8/9/21, 5:00 PM		Desarrollador Back-En...
70		Realizar programacion del modulo de seguridad	8/9/21, 8:00 AM	5 days	8/13/21, 5:00 PM		Desarrollador Back-En...
71		Realizar programacion del modulo de eventos	10/14/21, 9:00 AM	3 days?	10/19/21, 9:00 AM	81	Desarrollador Back-En...
72		Realizar programacion del modulo de inscripcion	10/4/21, 8:00 AM	6 days?	10/11/21, 5:00 PM	75	Desarrollador Back-En...
73		Realizar programación del modulo de recomendación	10/28/21, 10:00 AM	1 day?	10/29/21, 10:00 AM	79	Desarrollador Back-En...
74		Realizar programación del modulo de sharing	10/19/21, 11:00 AM	2 days	10/21/21, 11:00 AM	71	Desarrollador Back-En...
75		Realizar programacion del modulo de perfil	8/30/21, 8:00 AM	25 days?	10/1/21, 5:00 PM	76	Desarrollador Back-En...
76		Realizar programacion del modulo de parametrizacion	8/16/21, 8:00 AM	10 days	8/27/21, 5:00 PM	70	Desarrollador Back-En...
77		Realizar reunion con partes interesadas	9/13/21, 8:00 AM	1 day	9/13/21, 5:00 PM		Product owner
78		Realizar reunion con partes interesadas	10/11/21, 8:00 AM	1 day	10/11/21, 5:00 PM		Product owner
79		Realizar programacion del modulo de feedback	10/26/21, 3:00 PM	1.5 days?	10/28/21, 10:00 AM	80	Desarrollador Back-En...
80		Realizar programacion del modulo de pagos	10/21/21, 3:00 PM	3 days	10/26/21, 3:00 PM	74	Desarrollador Back-En...
81		Realizar programacion del modulo de publicidad	10/12/21, 8:00 AM	2 days?	10/13/21, 5:00 PM	72	Desarrollador Back-En...
82		Realizar programacion del modulo de reportes	10/29/21, 10:00 AM	2 days?	11/2/21, 10:00 AM	73	
83		Realizar pruebas de unidad	8/16/21, 8:00 AM	55 days	10/29/21, 5:00 PM	70	QA
84		Realizar pruebas de integración	8/30/21, 8:00 AM	28 days	10/6/21, 5:00 PM	76	QA

		Name	Start	Duration	Finish	Predecessors	Resource Names
85		Realizar pruebas de estrés	10/4/21, 8:00 AM	3 days	10/6/21, 5:00 PM	75	
86		Realizar pruebas de carga	10/7/21, 8:00 AM	4 days	10/12/21, 5:00 PM	84;85	QA
87		Realizar pruebas de seguridad por niveles de usuario	8/16/21, 8:00 AM	35 days	10/1/21, 5:00 PM	70	QA
88		Realizar documentación del sistema	10/4/21, 8:00 AM	12 days	10/19/21, 5:00 PM	75	Analista 1
89		Realizar manual de usuario	10/20/21, 8:00 AM	9 days?	11/1/21, 5:00 PM	88	Analista 1
90		Realizar implementacion del sistema	10/14/21, 8:00 AM	5 days	10/20/21, 5:00 PM	81	DevOps
91		Realizar capacitacion de usuarios	10/18/21, 1:00 PM	5 days	10/25/21, 1:00 PM		Analista 1;Product ow...
92		ENTREGABLES: Demo Aplicación, Manual de Usuario, Documentación "TP 1" en car...	11/2/21, 8:00 AM	0 days	11/2/21, 8:00 AM		
93		<b>Entregas</b>	<b>3/23/21, 8:00 AM</b>	<b>170 days</b>	<b>11/16/21, 8:00 AM</b>		
94		Empresa y Sistema	3/23/21, 8:00 AM	0 days	3/23/21, 8:00 AM		
95		Etapa de Definición de Requerimientos	4/27/21, 8:00 AM	0 days	4/27/21, 8:00 AM		
96		Etapa de Diseño	6/8/21, 8:00 AM	0 days	6/8/21, 8:00 AM		
97		Inicio de diseño de papers para Congreso CONAISI	6/15/21, 8:00 AM	0 days	6/15/21, 8:00 AM		
98		Demo de cada Sistema en aula para todo el curso	9/14/21, 8:00 AM	0 days	9/14/21, 8:00 AM		
99		Primera revisión de cada poster para exposición	9/28/21, 8:00 AM	0 days	9/28/21, 8:00 AM		
100		Segunda revisión de cada poster para exposición	10/12/21, 8:00 AM	0 days	10/12/21, 8:00 AM		
101		Demo de cada Sistema y poster para exposición	10/12/21, 8:00 AM	0 days	10/12/21, 8:00 AM		
102		Etapa de Desarrollo e Implementación	11/2/21, 8:00 AM	0 days	11/2/21, 8:00 AM		
103		Demo de cada Sistema y ensayo de exposición	11/9/21, 8:00 AM	0 days	11/9/21, 8:00 AM		
104		"15ª Exposición Anual de Proyectos de Sistemas	11/16/21, 8:00 AM	0 days	11/16/21, 8:00 AM		





25 Jul 21	1 Aug 21	8 Aug 21	15 Aug 21	22 Aug 21	29 Aug 21	5 Sep 21	12 Sep 21	19 Sep 21	26 Sep 21	3 Oct 21	10 Oct 21	17 Oct 21	24 Oct 21	31 Oct 21	7 Nov 21	14 Nov 21	21 Nov 21	28 Nov 21	5 Dec 21	12 Dec 21
F S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--





28 Feb 21	7 Mar 21	14 Mar 21	21 Mar 21	28 Mar 21	4 Apr 21	11 Apr 21	18 Apr 21	25 Apr 21	2 May 21	9 May 21	16 May 21	23 May 21	30 May 21	6 Jun 21	13 Jun 21	20 Jun 21	27 Jun 21	4 Jul 21	11 Jul 21	18 Jul 21
F S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F	S S M T W T F

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

25 Jul 21	1 Aug 21	8 Aug 21	15 Aug 21	22 Aug 21	29 Aug 21	5 Sep 21	12 Sep 21	19 Sep 21	26 Sep 21	3 Oct 21	10 Oct 21	17 Oct 21	24 Oct 21	31 Oct 21	7 Nov 21	14 Nov 21	21 Nov 21	28 Nov 21	5 Dec 21	12 Dec 21
F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F	F S S M T W T F

◆ 11/9

◆ 11/16

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--







# Proyecto Final

Ingeniería en Sistemas de Información 2021

## ANEXO 4

“Manual de usuario del sistema”





<b>Manual de Usuario del sistema</b>	<b>2</b>
Sistema y breve descripción	7
Objetivos	7
Principales beneficios	7
Requisitos de instalación	8
Forma de instalación	8
Descripción de todas las pantallas	8
Funcionalidades compartidas para todos los usuarios	8
Acceder al sistema	8
Acceder al sistema como administrador	9
Contactarse con el administrador del sistema	19
Editando el perfil del usuario	20
Visualización de árbitros en un evento	30
Cambio de contraseña y de email	30
Realizar feedback	34
Darse de baja del sistema	35
Modificar redes sociales	36
Agregar o quitar rol a usuario	36
Menús laterales	37
Sidebar administrador	38
Cargar publicidades	38
Roles especiales	39
Backups	42
Realización del backup	43
Restauración del backup	44
Listado de los tipos de organizadores	46
Creación y modificación de los tipos de organizadores	47
Visualización de los deportes	48
Visualización de países, provincias, departamentos y localidades	49
Creación de países, provincias, departamentos y localidades.	49
Sidebar deportista	52
Visualización de equipos	52
Modificación y alta de equipos	53
Inscripción a eventos	53
Crear deporte favorito	55
Sidebar organizador	56
Visualización de las categorías	57
Modificación y alta de las categorías	57
Visualización de eventos de un organizador	58
Creación de eventos	58
Organizar Eventos	62
Agregar partidos	63

Agregar Árbitros	64
Eliminar Árbitros	64
Eliminar Equipos	64
Registrar pagos	65
Ver solicitudes de patrocinio	66
Visualizar brackets	66
Registrar Resultado de un partido	67
Tipos de organizadores	67
Vincular cuenta con mercado pago	68
Reportes	70
Tipos de deportes usados para eventos	70
Cantidad de inscriptos por mes	71
Sidebar patrocinador	72
Visualización de solicitudes de patrocinio	73
Visualización de eventos patrocinados	73
Sidebar árbitro	74
Aceptar invitación de arbitraje por mail	75
Visualizar eventos aceptados	76
Visualizar eventos aceptados	77
Sidebar club	77
Visualización de canchas	78
Creación y modificación de canchas	79
Visualización de equipos	80
Creación y modificación de equipos	80
Ver solicitudes de creación de eventos pendientes	81
Manejo y visualización de errores	82
<b>Índice de figuras</b>	<b>3</b>
Figura 1: Pantalla de inicio de sesión	9
Figura 2: Inicio de sesión como un usuario administrador.	10
Figura 3: Pantalla de registro	11
Figura 4: Pantalla de registro para los deportistas	12
Figura 5: Pantalla de campos adicionales para el registro de los organizadores	13
Figura 6: Pantalla de registro para los árbitros	14
Figura 7: Pantalla de registro para un club	16
Figura 8: Pantalla de registro para un patrocinador	17
Figura 9: Pantalla de visualización de eventos	18
Figura 10: Pantalla de visualización de eventos con un filtro determinado	18
Figura 11: Pantalla de visualización de eventos sin ningún tipo de filtro	19
Figura 12: Imagen que nos muestra la cantidad de ítems por página y la actual	19
Figura 13: Imagen que nos muestra la pantalla de contacto con el administrador	20

Figura 14: Pestaña de acceso a configuración del perfil	20
Figura 15: Pantalla que nos muestra la pantalla principal del perfil del deportista	21
Figura 16: Pantalla que nos muestra la pantalla principal del perfil del organizador	22
Figura 17: Pantalla que nos muestra la pantalla principal del perfil del árbitro	23
Figura 18: Pantalla que nos muestra la pantalla principal del perfil del club	23
Figura 19: Pantalla que nos muestra la pantalla principal del perfil del patrocinador	24
Figura 20: Pantalla que nos muestra como editar el perfil desde la perspectiva del patrocinador.	25
Figura 21: Pantalla que nos muestra como editar el perfil desde la perspectiva del deportista.	26
Figura 22: Pantalla que nos muestra como editar el perfil desde la perspectiva del organizador.	26
Figura 23: Pantalla que nos muestra como editar el perfil desde la perspectiva del club.	27
Figura 24: Imagen que nos muestra cómo cambiar la foto de perfil.	28
Figura 25: Pantalla que nos muestra características de un evento en particular.	28
Figura 26: Pantalla que nos muestra la categoría de un evento en particular.	29
Figura 27: Pantalla con portada de evento.	30
Figura 28: Ventana de visualización de árbitros de un evento.	30
Figura 29: Pantalla que nos muestra cómo cambiar la contraseña.	31
Figura 30: Pantalla de confirmación de contraseña.	32
Figura 31: Pantalla de recuperación de contraseña.	32
Figura 32: Pantalla del email enviado para cambiar la contraseña.	32
Figura 33: Pantalla de solicitud de cambio de contraseña.	33
Figura 34: Pantalla de cambio de email.	33
Figura 35: Pantalla de confirmación de cambio de email	34
Figura 36: Botón de feedback en categoría de evento	34
Figura 37: Ventana de feedback	35
Figura 38: Confirmación para darse de baja del sistema	36
Figura 39: Confirmación para darse de baja del sistema	36
Figura 40: Agregar rol a usuario	37
Figura 41: Quitar rol a usuario	37
Figura 42: Sidebar del administrador	38
Figura 43: Pantalla de carga de publicidad en la aplicación	39
Figura 44: Pantalla que muestra la publicidad a lo largo del sistema	39
Figura 45: Visualización de los roles en el sistema	40
Figura 46: Creación de roles en el sistema	40
Figura 47: Modificación de un rol en el sistema	41
Figura 48: Visualización de los usuarios con roles especiales	41

Figura 49: Asignación de roles especiales por parte del administrador	42
Figura 50: Apartado de backup en el sidebar del administrador	42
Figura 51: Botón para realizar un backup	42
Figura 52: Botón para restaurar un backup	43
Figura 53: Mensaje de realización de backup en curso	43
Figura 54: Mensaje de backup realizado correctamente	43
Figura 55: Imagen que nos muestra la ubicación de guardado del backup	44
Figura 56: Imagen que nos muestra el backup guardado correctamente	44
Figura 57: Selección del backup.	45
Figura 58: Mensaje de error a la hora de seleccionar el backup.	45
Figura 59: Mensaje de procesamiento del backup	45
Figura 60: Mensaje de éxito de restauración del backup.	46
Figura 61: Funcionamiento del backup	46
Figura 62: Listado de los tipos de organizadores en el sistema	47
Figura 63: Creación de un tipo organizador en el sistema	47
Figura 64: Modificación de un tipo organizador en el sistema	48
Figura 65: Pantalla de visualización de los deportes	48
Figura 66: Pantalla de visualización de las provincias	49
Figura 67: Creación del país	50
Figura 68: Creación de la provincia	50
Figura 69: Creación del departamento	51
Figura 70: Creación de la localidad	51
Figura 71: Sidebar del deportista	52
Figura 72: Pantalla de visualización de equipos	52
Figura 73: Pantalla de alta y modificación de equipos	53
Figura 74: Pantalla de eventos, sección categorías de un evento	54
Figura 75: Pantalla de categoría de un evento	55
Figura 76: Pantalla de inscripción a categoría de un evento	55
Figura 77: Creación de deporte favorito	56
Figura 78: Sidebar del organizador	56
Figura 79: Pantalla de visualización de las categorías.	57
Figura 80: Pantalla de modificación de las categorías.	58
Figura 81: Pantalla que nos muestra todos los eventos del sistema.	58
Figura 82: Pantalla que nos muestra el inicio de la creación de un evento	59
Figura 83: Pantalla para agregar categorías al evento	60
Figura 84: Pantalla que nos muestra la creación de una categoría del evento	61
Figura 85: Pantalla de visualización de los clubes de la jornada.	62
Figura 86: Pantalla de organización de evento	63
Figura 87: Pantalla de agregar partido a evento	63
Figura 88: Ventana de invitación de árbitros al evento	64
Figura 89: Ventana para eliminar árbitros del evento	64
Figura 90: Ventana de eliminación de equipos del evento	65
Figura 91: Ventana de registro de pagos de equipos en un evento	66

Figura 92: Ventana de solicitudes entrantes de patrocinio	66
Figura 93: Sección de Brackets	66
Figura 94: Ventana para registrar resultado de un partido	67
Figura 95: Pantalla de visualización de cuentas de los organizadores	68
Figura 96: Pantalla de visualización de la integración con Mercado Pago	68
Figura 97: Apartado para vincular la cuenta de mercado pago como organizador	69
Figura 98: Ingreso a mercado pago con nuestra cuenta.	70
Figura 99: Pantalla de reporte de deportes usados para eventos	71
Figura 100: Pantalla de reporte de cantidad de inscriptos por mes	72
Figura 101: Sidebar del patrocinador	72
Figura 102: Pantalla de visualización de evento patrocinadores	72
Figura 103: Formulario de solicitud de patrocinio	73
Figura 104: Visualización de eventos pendientes de respuesta de patrocinio.	73
Figura 105: Visualización de eventos patrocinados actualmente	74
Figura 106: Sidebar del árbitro	75
Figura 107: Email de solicitud de participación como arbitro	75
Figura 108: Ventana de confirmación de participación en evento como árbitro	76
Figura 109: Visualizar eventos aceptados como árbitro	76
Figura 110: Visualizar eventos aceptados en curso como árbitro	77
Figura 111: Sidebar del club	78
Figura 112: Pestaña de visualización de canchas como club	78
Figura 113: Pantalla de visualización de canchas como club	79
Figura 114: Pantalla de creación de canchas como club	79
Figura 115: Pestaña de visualización de equipos como club	80
Figura 116: Visualización de equipos como club	80
Figura 117: Pantalla de creación de equipos como club	81
Figura 118: Solicitudes pendientes de club	82
Figura 119: Modal confirmar acción solicitud	82
Figura 120: Pantalla de visualización de errores de ingreso de datos	82
Figura 121: Pantalla de visualización de error de login incorrecto	83
Figura 122: Pantalla de visualización de error grave al querer editar evento finalizado.	84
Figura 123: Pantalla de contacto.	84

# Manual de Usuario del sistema

## Sistema y breve descripción

Sportinev es un sistema de gestión de eventos deportivos cuyo objetivo es desarrollar un sistema de información que permita a los organizadores crear eventos y llevar un registro de ellos, conectándose con los deportistas y patrocinadores interesados. El impacto consiste en generar una red de organizaciones que posibiliten a los deportistas del sistema tener una gran variedad de opciones, generando así mayor publicidad para los eventos disputados, esto quiere decir que las organizaciones podrán exponer su información hacia los clientes para que estos puedan verse interesados.

## Objetivos

El objetivo del sistema es proveer una herramienta que facilite la gestión de eventos deportivos y que motive a la generación de más eventos deportivos. Para esto, el sistema busca satisfacer las necesidades de todas las entidades que pueden participar en un evento deportivo, como pueden ser, el organizador, los deportistas, los patrocinadores y hasta incluso las necesidades de los árbitros del evento. Además, busca motivar a los deportistas a participar de los eventos mediante la posibilidad de estar entre los deportistas destacados de la aplicación y de ser observado por patrocinadores.

## Principales beneficios

Los principales beneficios de la realización del manual de usuario para el sistema son:

- Excelente gestión de los eventos deportivos por parte de las organizaciones.
- Mayor interés de los deportistas por participar en los eventos.
- Menor estrés por parte de los deportistas, ya que la búsqueda y participación de los eventos es mucho mejor.
- Mayor cantidad de deportistas inscriptos en los eventos, ya que los eventos tienen una mayor difusión.
- Menor utilización de papel.
- Una de las mayores ventajas del sistema y bienes más preciados para los usuarios, es la información disponible en él. Gracias a la información, los usuarios pueden tomar mejores decisiones respecto al ámbito de aplicación, y lograr una mayor satisfacción respecto a la usabilidad del sistema. Debido a esto existe el módulo reportes, que muestra varios tipos de reportes a los diferentes roles del sistema, con basta información sobre los usuarios deportistas, clubes, patrocinadores, árbitros y organizadores que forman parte del sistema o de un evento en particular. Puede ser mostrada de formas muy dinámicas para satisfacer las expectativas del usuario. Previamente, los organizadores debían cargar esta información en hojas de cálculo y

realizar un arduo trabajo matemático para mostrar los mismos gráficos y estadísticas que ofrece la aplicación en un solo clic. Ya no debe cargar manualmente estos datos y más importante aún, tiene la información de los demás eventos y organizadores a su alcance, que previamente era demasiado complicado, o en algunos casos imposible de conseguir.

## Requisitos de instalación

Para el correcto funcionamiento del sistema, el usuario deberá poseer una computadora o celular con los siguientes requerimientos mínimos:

En el caso de que el usuario a la aplicación desde una computadora personal o notebook:

- 2 GB de memoria RAM.
- Intel i3-2130
- Conexión a internet.
- Navegador web moderno instalado.

En el caso de que el usuario ingrese a la aplicación desde una tablet o dispositivo móvil:

- Pantalla con resolución superior a 1280x720px.
- Pantalla mínima de 10".
- Conexión a internet inalámbrica o datos.
- Android 4.0.4 o iOS 5.

## Forma de instalación

El usuario que utilizara el sistema no deberá instalar ningún tipo de software, ya que el sistema se encuentra basado en la web. Simplemente, deberá contener un navegador web moderno, como Google Chrome, Firefox o Brave.

## Descripción de todas las pantallas

Funcionalidades compartidas para todos los usuarios

Acceder al sistema

Para poder acceder al sistema el usuario deberá seleccionar el tipo de rol, su email y su contraseña.

Una vez completados todos los campos, debe presionar en el botón "ingresar", el cual si todos los datos son válidos y correctos lo dejará ingresar al sistema.

**Iniciar sesión**

Tipo de usuario  
DEPORTISTA

Ingrese su email  
miguelcorso@gmail.com

Ingrese su contraseña  
.....

[¿Olvidaste tu contraseña?](#)

Ingresar

Volver

**¿No sos miembro?**

Regístrate

Figura 1: Pantalla de inicio de sesión

Acceder al sistema como administrador

Para que un administrador pueda iniciar sesión lo primero que deberá realizar es ingresar a la ruta oculta llamada /login/administrador/, dicha ruta se encuentra oculta por cuestiones de seguridad e integridad de la aplicación.

En tipo de usuario seleccionara el de administrador del menú desplegable e ingresara con sus credenciales.





The image shows a login interface for 'SPORTINEV'. At the top, there is a logo consisting of two hands shaking in blue and red, followed by the text 'SPORTINEV'. Below the logo, the title 'Iniciar sesión' is displayed. The form includes a dropdown menu for 'Tipo de usuario' with 'ADMINISTRADOR' selected. There are input fields for 'Ingrese su email' (containing 'miguelcorso@gmail.com') and 'Ingrese su contraseña' (with masked characters). A '¿Olvidaste tu contraseña?' link is visible. At the bottom, there are two buttons: a prominent red 'Ingresar' button and a smaller 'Volver' button.

Figura 2: Inicio de sesión como un usuario administrador.

En caso de que el usuario no tenga una cuenta, deberá crearse una, en la cual se le pedirá los siguientes datos básicos:

- nombre
- teléfono
- email
- contraseña
- tipo de usuario

**SPORTINEV**

## Registrarse

Ingrese su nombre  
Pepe

Ingrese su teléfono o celular  
2616104635

Ingrese su email  
miguelcorso@gmail.com

Ingrese su contraseña  
.....

Tipo de usuario

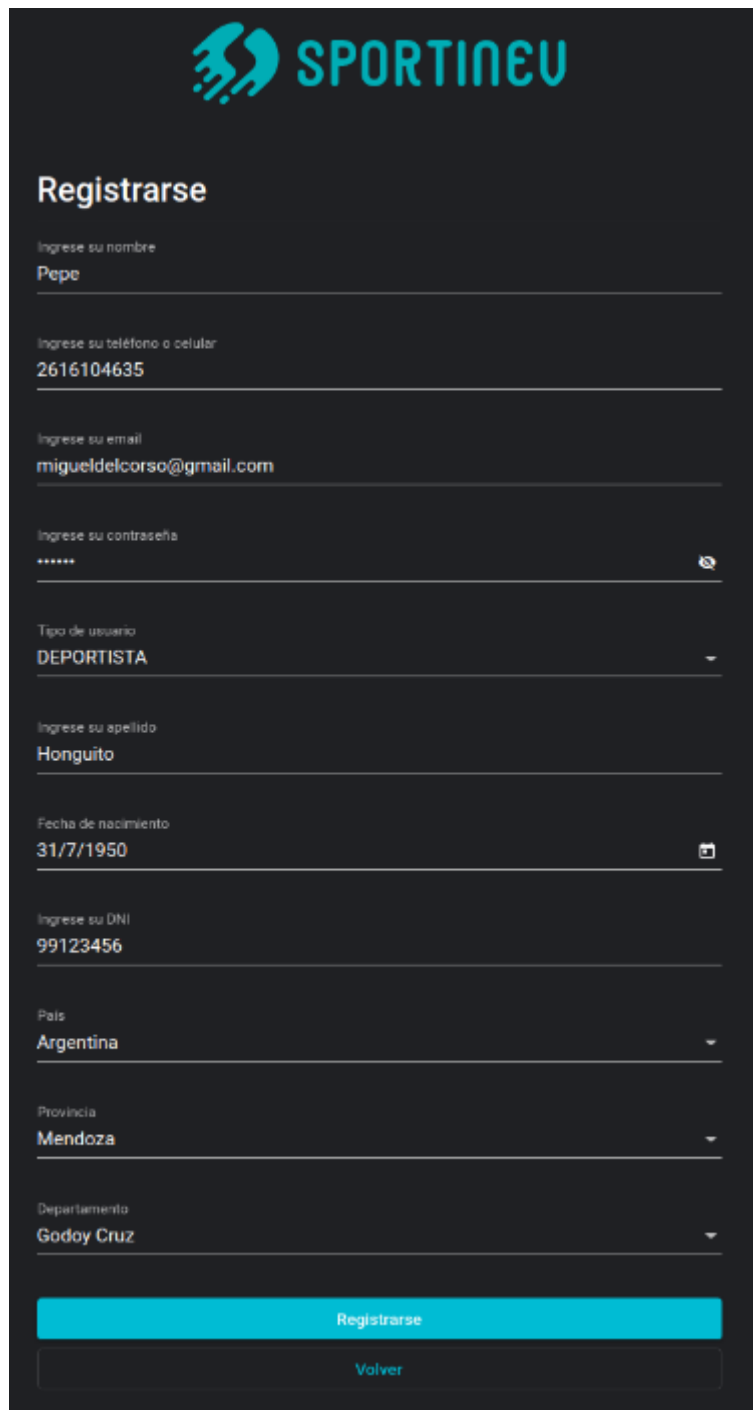
Registrarse

Volver

Figura 3: Pantalla de registro

En caso de que el tipo de usuario sea deportista, se deberá proveer adicionalmente los siguientes campos:

- apellido
- fecha de nacimiento
- DNI
- país
- provincia
- departamento



The image shows a registration form for 'SPORTINEV'. The form is titled 'Registrarse' and contains the following fields and options:

- Ingrese su nombre:** Pepe
- Ingrese su teléfono o celular:** 2616104635
- Ingrese su email:** migueldelcorso@gmail.com
- Ingrese su contraseña:** \*\*\*\*\*
- Tipo de usuario:** DEPORTISTA (dropdown menu)
- Ingrese su apellido:** Honguito
- Fecha de nacimiento:** 31/7/1950
- Ingrese su DNI:** 99123456
- País:** Argentina (dropdown menu)
- Provincia:** Mendoza (dropdown menu)
- Departamento:** Godoy Cruz (dropdown menu)

At the bottom of the form, there are two buttons: a red 'Registrarse' button and a grey 'Volver' button.

Figura 4: Pantalla de registro para los deportistas

En caso de que el usuario se quiera registrar como organizador, se deberá proveer adicionalmente los siguientes campos:

- CUIT
- País

- Provincia

Ingrese su CUIT  
27280335148

País  
Argentina

Provincia  
Mendoza

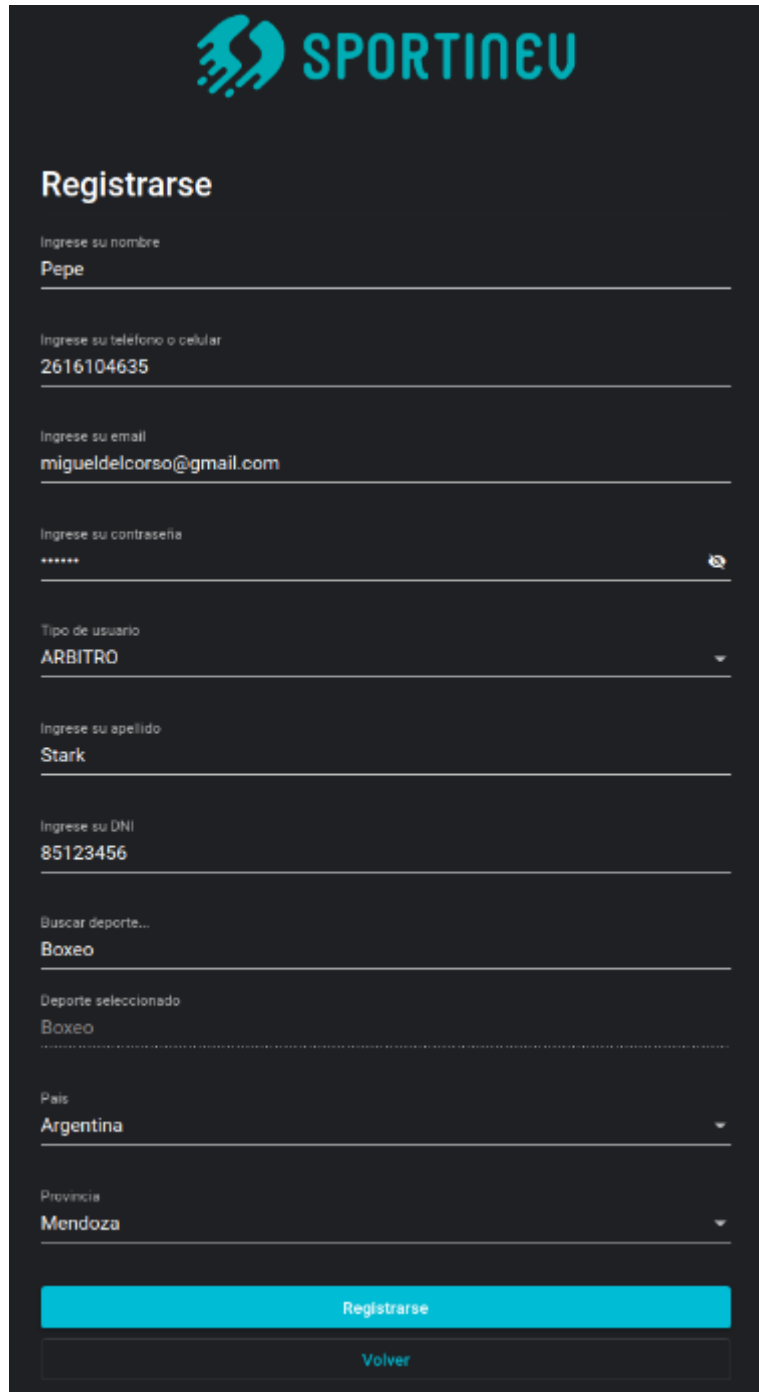
Registrarse

Volver

Figura 5: Pantalla de campos adicionales para el registro de los organizadores

En caso de que el usuario se quiera registrar como árbitro, se deberá proveer adicionalmente los siguientes campos:

- Apellido
- DNI
- País
- Provincia



The image shows a registration form for referees on the Sportinev platform. The form is titled 'Registrarse' and includes the following fields and options:

- Ingrese su nombre:** Pepe
- Ingrese su telefono o celular:** 2616104635
- Ingrese su email:** miguelcorso@gmail.com
- Ingrese su contraseña:** \*\*\*\*\*
- Tipo de usuario:** ARBITRO
- Ingrese su apellido:** Stark
- Ingrese su DNI:** 85123456
- Buscar deporte...:** Boxeo
- Deporte seleccionado:** Boxeo
- País:** Argentina
- Provincia:** Mendoza

At the bottom of the form, there are two buttons: 'Registrarse' (highlighted in red) and 'Volver'.

Figura 6: Pantalla de registro para los árbitros

En caso de que el usuario se quiera registrar como club, se deberá proveer adicionalmente los siguientes campos:

- Domicilio
- CUIT
- País
- Provincia
- Departamento

- Localidad

Además, se le proveerá de un muy conveniente mapa que le permite seleccionar de forma rápida y sencilla la ubicación del club.

CLUB

Ingrese su domicilio  
Godoy Cruz

Ingrese su CUIT  
20405609987

País  
Argentina


Provincia  
Mendoza

Departamento  
Godoy Cruz

Localidad  
GODOY CRUZ

### Seleccione la ubicación del club

Buscar...  
Godoy Cruz

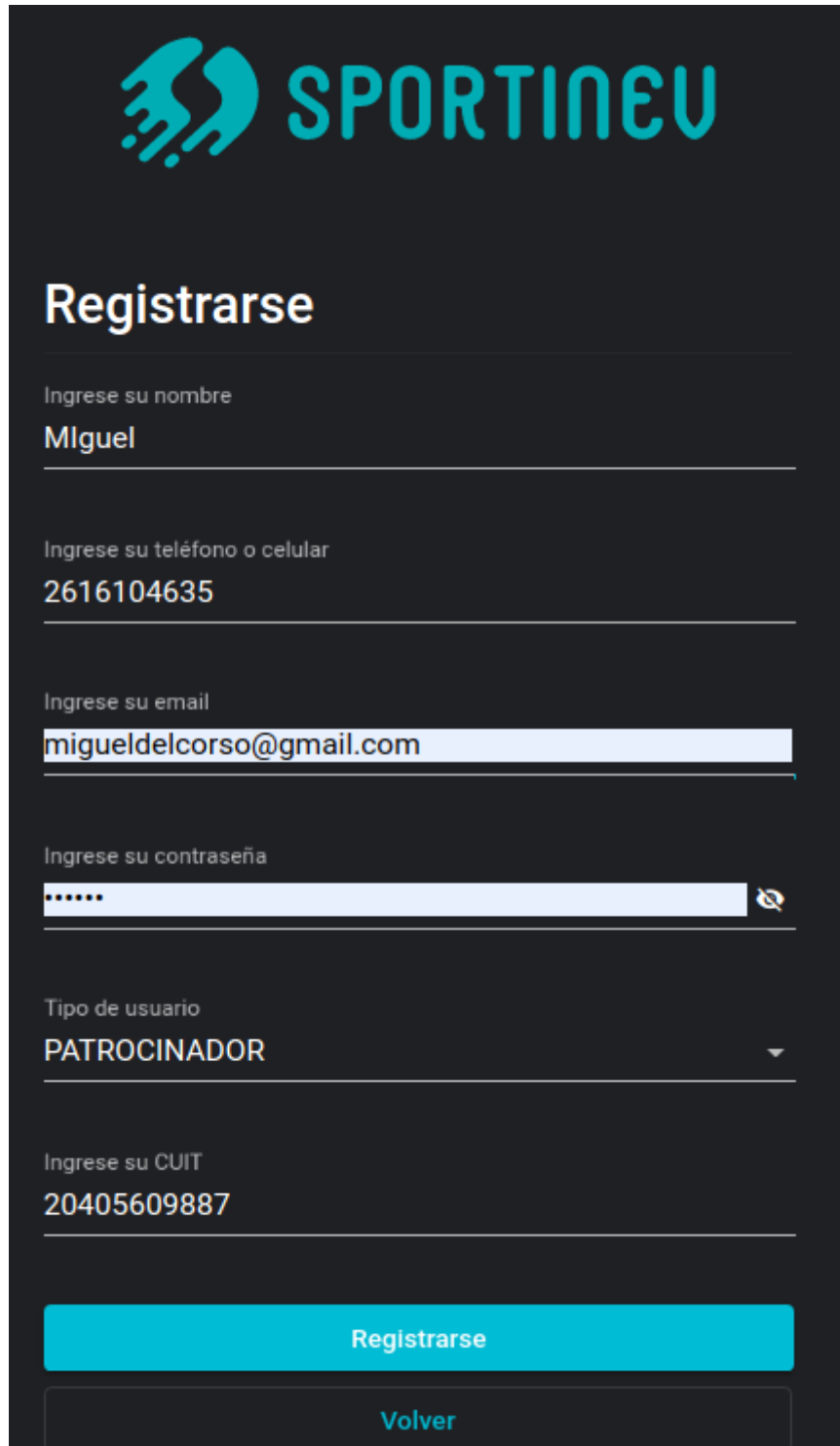


Registrarse

Volver

Figura 7: Pantalla de registro para un club

Finalmente en caso de que se desea registrar como un patrocinador deberá proveer un apellido y CUIL válido.



The image shows a registration form for Sportinev. At the top, there is a logo consisting of a stylized blue hand icon and the word "SPORTINEV" in blue capital letters. Below the logo, the title "Registrarse" is displayed in white. The form contains several input fields: "Ingrese su nombre" with the value "Miguel"; "Ingrese su teléfono o celular" with the value "2616104635"; "Ingrese su email" with the value "migueldelcorso@gmail.com"; "Ingrese su contraseña" with a masked password "....." and a toggle icon; "Tipo de usuario" with a dropdown menu showing "PATROCINADOR"; and "Ingrese su CUIT" with the value "20405609887". At the bottom, there are two buttons: a prominent blue "Registrarse" button and a white "Volver" button.

Figura 8: Pantalla de registro para un patrocinador

Visualizar eventos de pantalla principal

El usuario puede visualizar los eventos, que la aplicación se encarga de filtrar de acuerdo a una serie de algoritmos para solo mostrar los eventos en los que puede llegar a estar interesado.

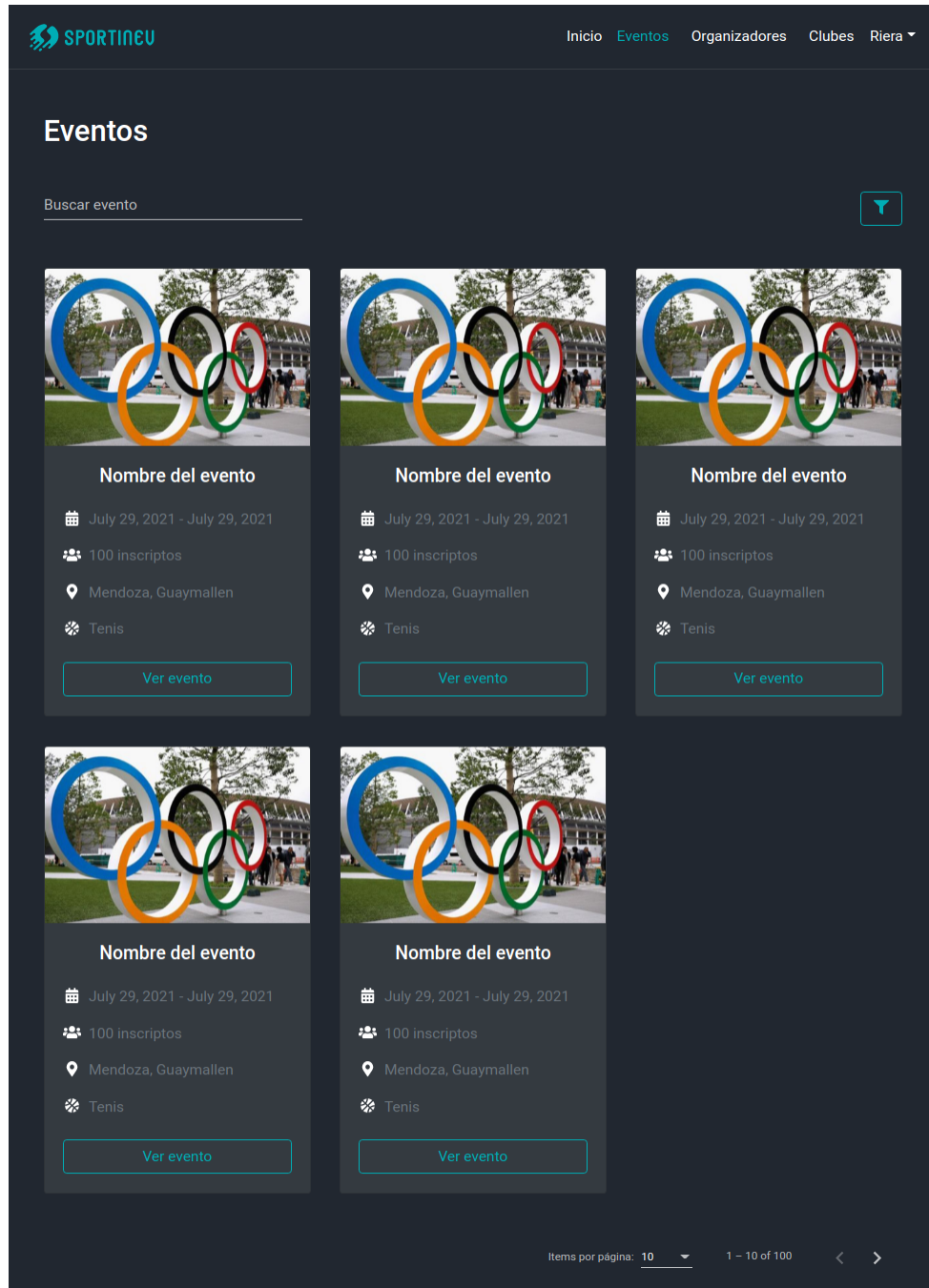




Figura 9: Pantalla de visualización de eventos

Además, el usuario cuenta con la posibilidad de filtrar aún más los resultados, para lo cual podrá buscar en la barra de búsqueda provista o bien utilizar algunos de los filtros especializados.

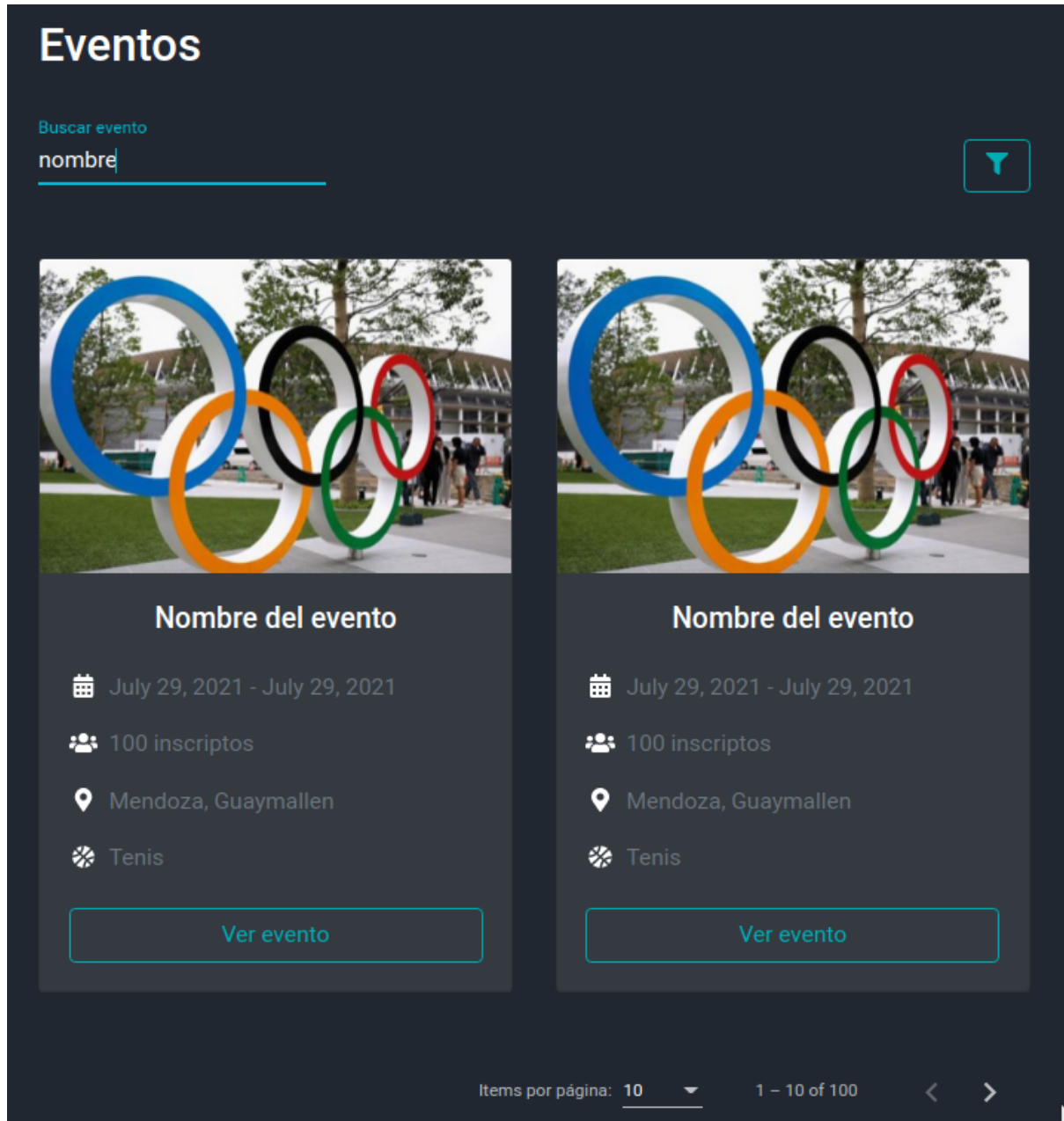


Figura 10: Pantalla de visualización de eventos con un filtro determinado

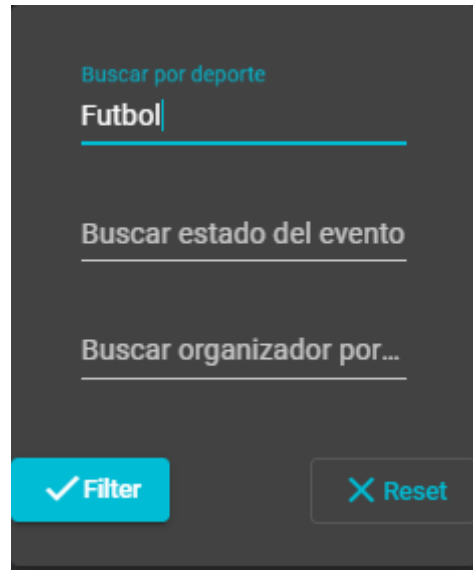


Figura 11: Pantalla de visualización de eventos sin ningún tipo de filtro

En la parte final de la pantalla el usuario podrá manejar la cantidad de eventos que se van a mostrar por página, además tiene la posibilidad de pasar a la siguiente página.



Figura 12: Imagen que nos muestra la cantidad de ítems por página y la actual

Contactarse con el administrador del sistema

Un usuario se puede contactar con el administrador del sistema para poder informarle sobre errores que se han producido al usar la aplicación o pedirle nuevas funcionalidades. Una vez enviado el mensaje el usuario tendrá que esperar 4 horas para poder volver a enviar otro, esto se realiza a fin de evitar spam. Los campos solicitados son:

- Nombre
- Apellido
- Teléfono
- email
- Comentario

SPORTINEV

Por favor complete el siguiente cuestionario para que nuestros analistas lo revisen

Ingrese su nombre  
Armando

Ingrese su apellido  
Casas

Ingrese su teléfono  
2616104635

Ingrese su email  
armandocasas@gmail.com

Mensaje para administrador  
Solo quería comentar que es la mejor aplicación de eventos deportivos.

Registrarse

Volver

Figura 13: Imagen que nos muestra la pantalla de contacto con el administrador

Editando el perfil del usuario

Una vez que el usuario haya iniciado correctamente sesión en la parte superior de la pantalla tendrá un apartado con su nombre, el cual al hacer clic y dirigirse a la sección de perfil podrá mirar más información útil de acuerdo al rol con que ingresó al sistema.

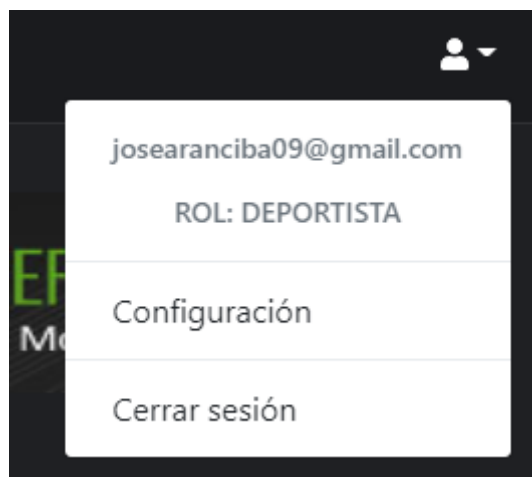


Figura 14: Pestaña de acceso a configuración del perfil

En las próximas imágenes se muestra como luce el perfil para uno de los roles del sistema.



Figura 15: Pantalla que nos muestra la pantalla principal del perfil del deportista



Figura 16: Pantalla que nos muestra la pantalla principal del perfil del organizador



Figura 17: Pantalla que nos muestra la pantalla principal del perfil del árbitro



Figura 18: Pantalla que nos muestra la pantalla principal del perfil del club




Figura 19: Pantalla que nos muestra la pantalla principal del perfil del patrocinador

En caso de que el usuario desee actualizar sus datos personales, simplemente hará clic en "Actualizar datos". Dependiendo del rol van a ser los datos que se le soliciten.

Para el caso de los patrocinadores solamente se le solicitará el nombre, teléfono celular y CUIT, como se muestra en la figura.

## Editar datos



Ingrese su nombre  
jose

Ingrese su teléfono o celular  
2644638015

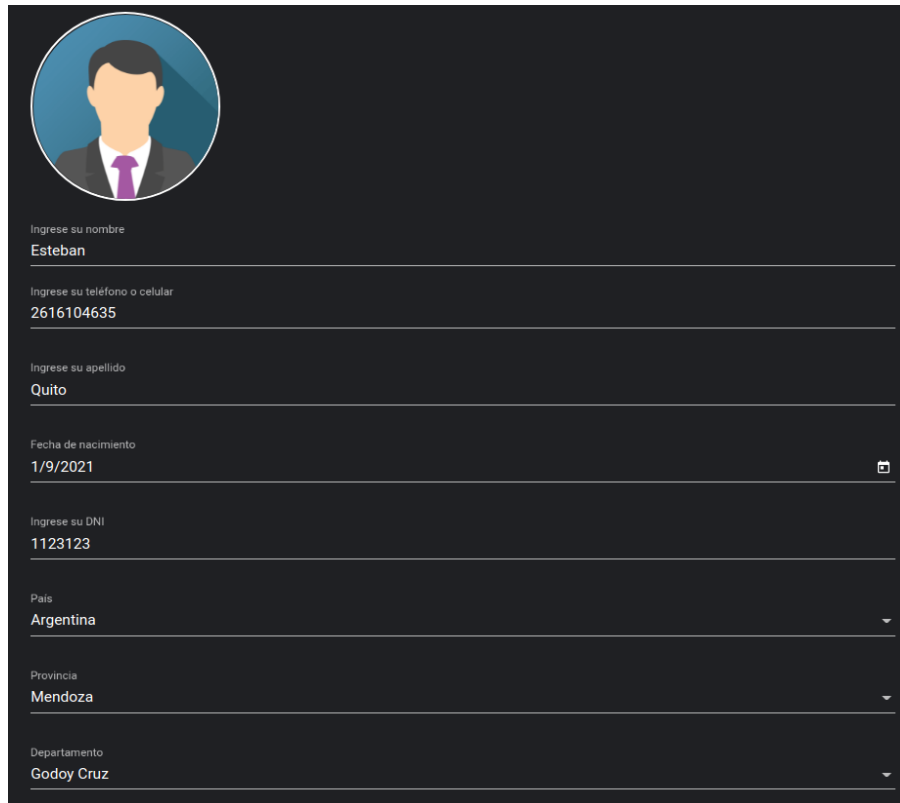
Ingrese su CUIT  
20407795815

[Volver](#) [Guardar](#)

Figura 20: Pantalla que nos muestra como editar el perfil desde la perspectiva del patrocinador.

En caso de haber ingresado como deportista se le solicitarán los siguientes datos:

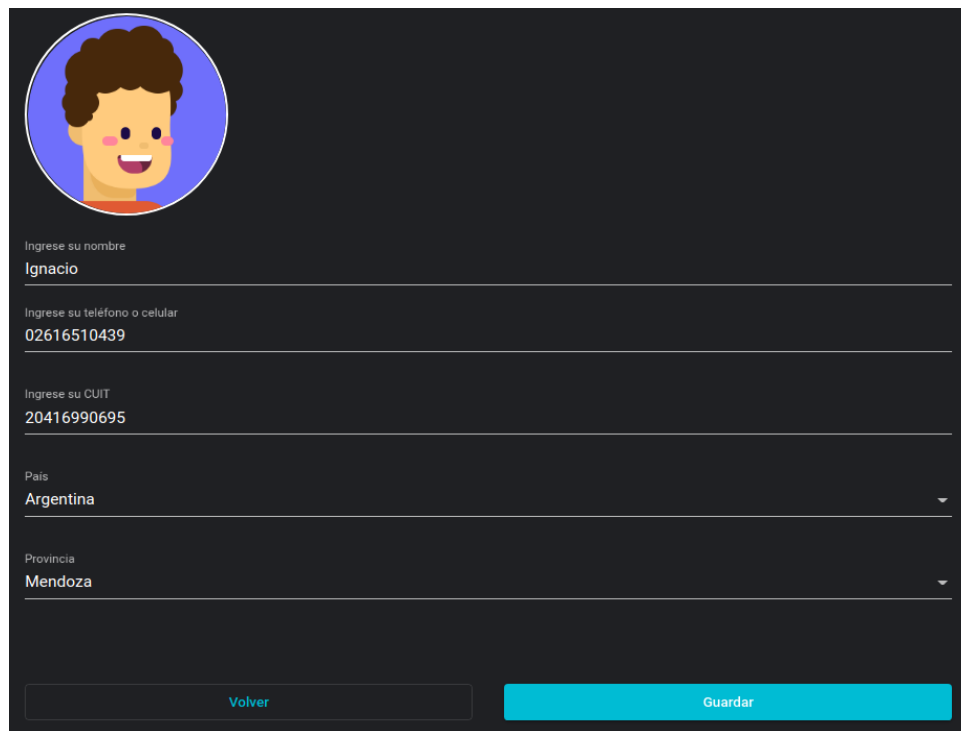




A screenshot of a user profile editing interface. At the top left is a circular profile picture of a man in a suit. Below it are several input fields with labels and values:

- Ingrese su nombre:** Esteban
- Ingrese su teléfono o celular:** 2616104635
- Ingrese su apellido:** Quito
- Fecha de nacimiento:** 1/9/2021
- Ingrese su DNI:** 1123123
- Pais:** Argentina
- Provincia:** Mendoza
- Departamento:** Godoy Cruz

Figura 21: Pantalla que nos muestra como editar el perfil desde la perspectiva del deportista.



A screenshot of a user profile editing interface. At the top left is a circular profile picture of a smiling woman. Below it are several input fields with labels and values:

- Ingrese su nombre:** Ignacio
- Ingrese su teléfono o celular:** 02616510439
- Ingrese su CUIT:** 20416990695
- Pais:** Argentina
- Provincia:** Mendoza

At the bottom of the form are two buttons: "Volver" (left) and "Guardar" (right).

Figura 22: Pantalla que nos muestra como editar el perfil desde la perspectiva del organizador.



Ingrese su nombre  
Ignacio

Ingrese su teléfono o celular  
02616510467

Ingrese su apellido  
Stark

Ingrese su DNI  
41699065

Buscar deporte...

Deporte seleccionado  
Futbol

País

Provincia

Figura 23: Pantalla que nos muestra como editar el perfil desde la perspectiva del club.

Una de las características comunes al editar el perfil del usuario es que se puede cambiar la imagen del mismo, para poder realizar esto el usuario se posiciona sobre su imagen actual y hará clic en la misma, donde aparecerá un diálogo para poder subir una imagen desde su computadora.

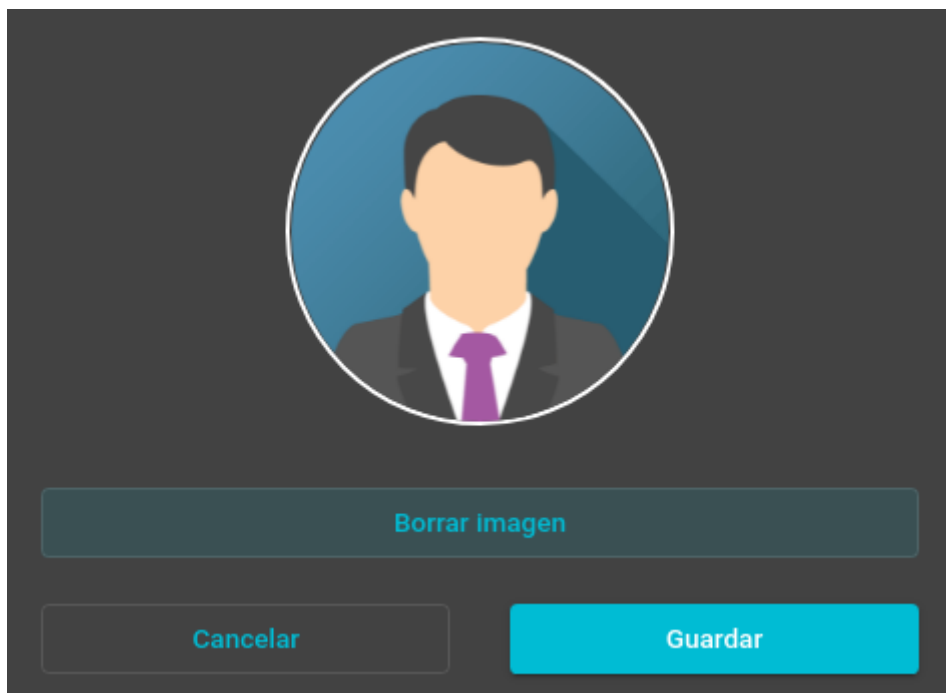


Figura 24: Imagen que nos muestra cómo cambiar la foto de perfil.

Visualización de un evento

Cualquier usuario del sistema puede visualizar un evento en particular del sistema como se muestra en la siguiente figura:

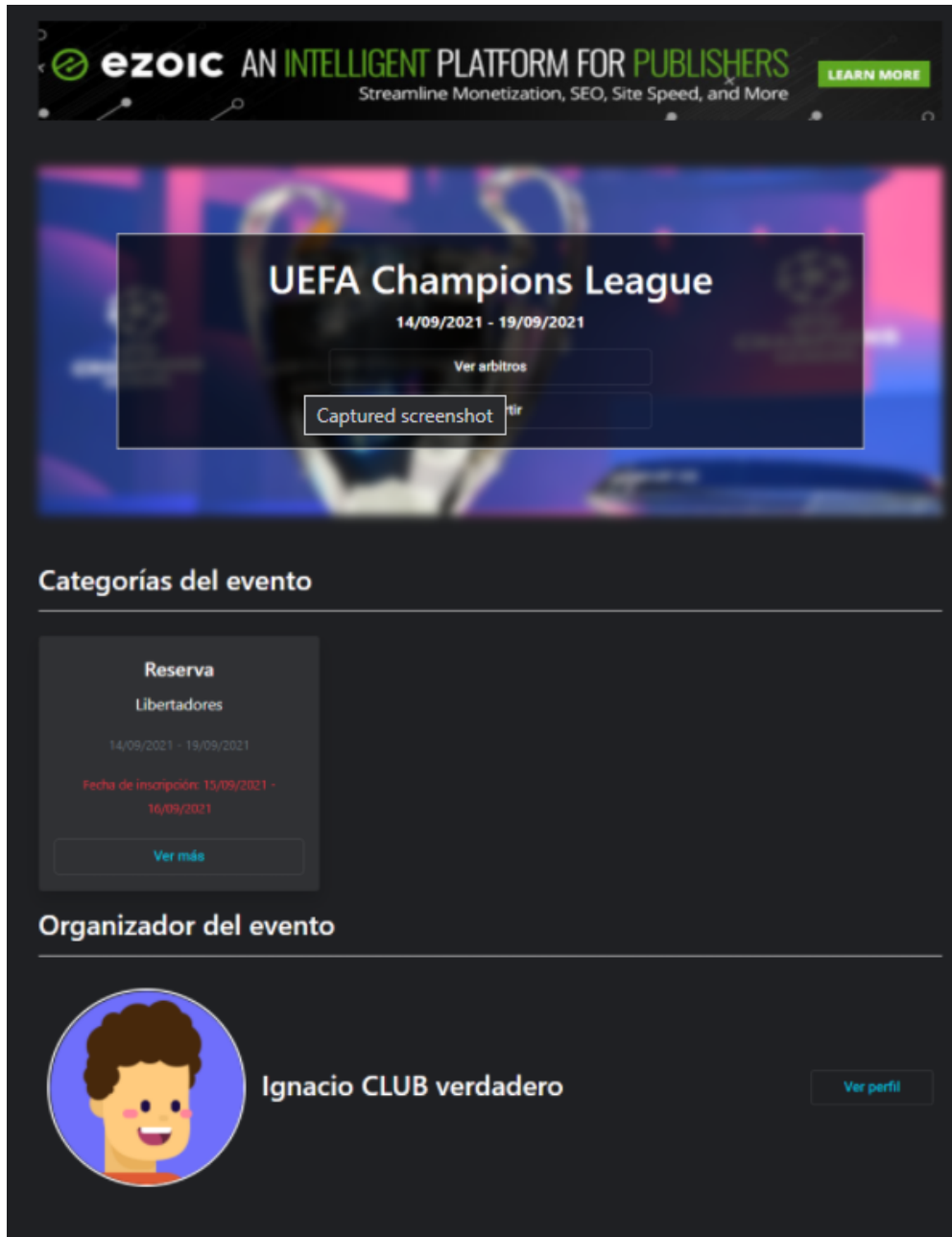


Figura 25. Pantalla que nos muestra características de un evento en particular.

Como se observa se muestra:

- El nombre del evento
- La imagen del evento

- Botones para compartir el evento en las redes sociales.
- Las categorías que tiene ese evento con toda su información correspondiente
- Quien es el organizador del evento y su respectiva imagen

Si el usuario hace clic en el apartado de “ver más” en una categoría, podremos observar detalles de la misma, como lo indica la siguiente figura:

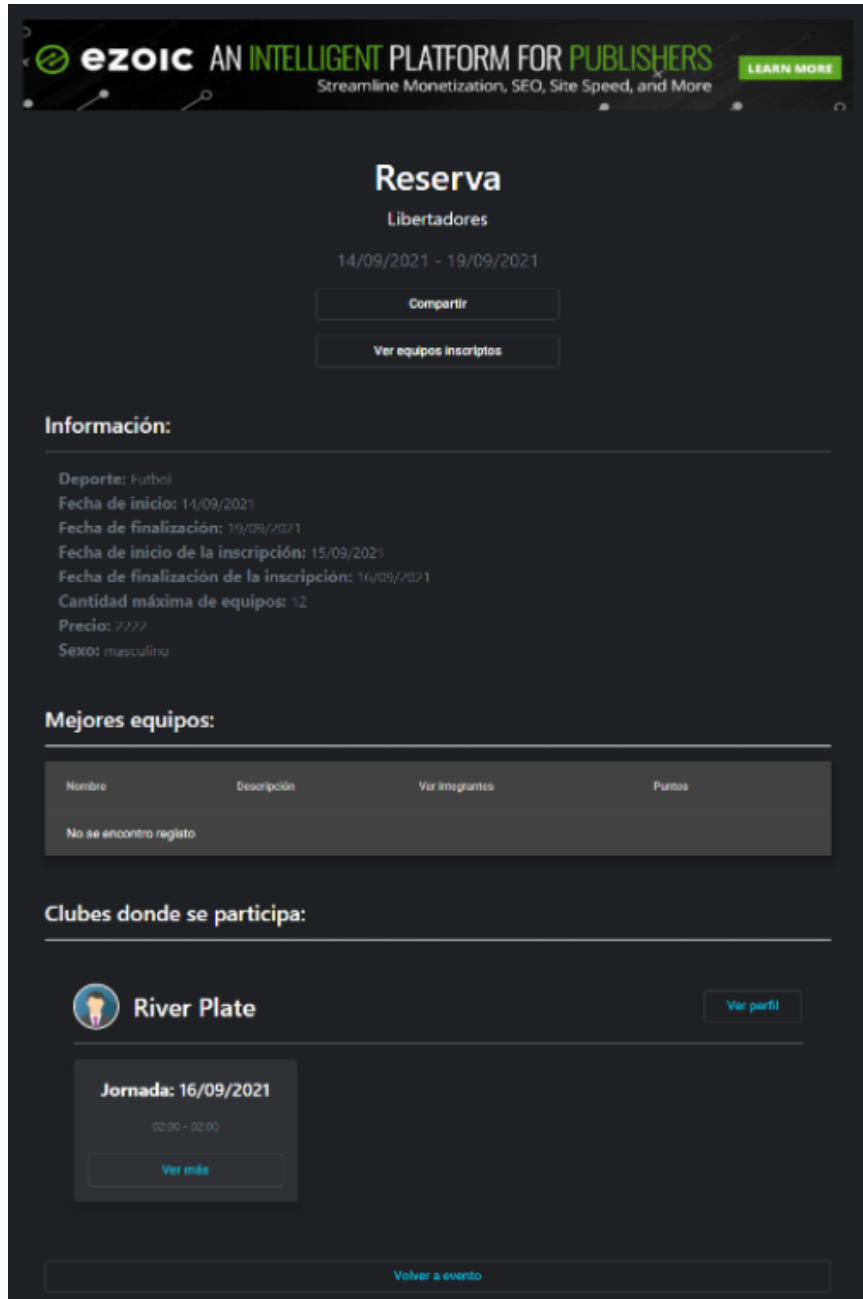


Figura 26: Pantalla que nos muestra la categoría de un evento en particular.

Cabe mencionar que los mejores equipos del evento serán mostrados una vez que el evento finalizado, mostrando los 3 mejores equipos del evento.

### Visualización de árbitros en un evento

Para visualizar los árbitros de un evento, un usuario correctamente logueado en la aplicación puede ingresar al evento en cuestión, y en la portada del evento aparecerá el botón de “Ver árbitros”.



Figura 27: Pantalla con portada de evento.

Una vez clickeado el botón “Ver árbitros”, el sistema le muestra al usuario una ventana con una tabla que contiene los árbitros que participarán del evento seleccionado.

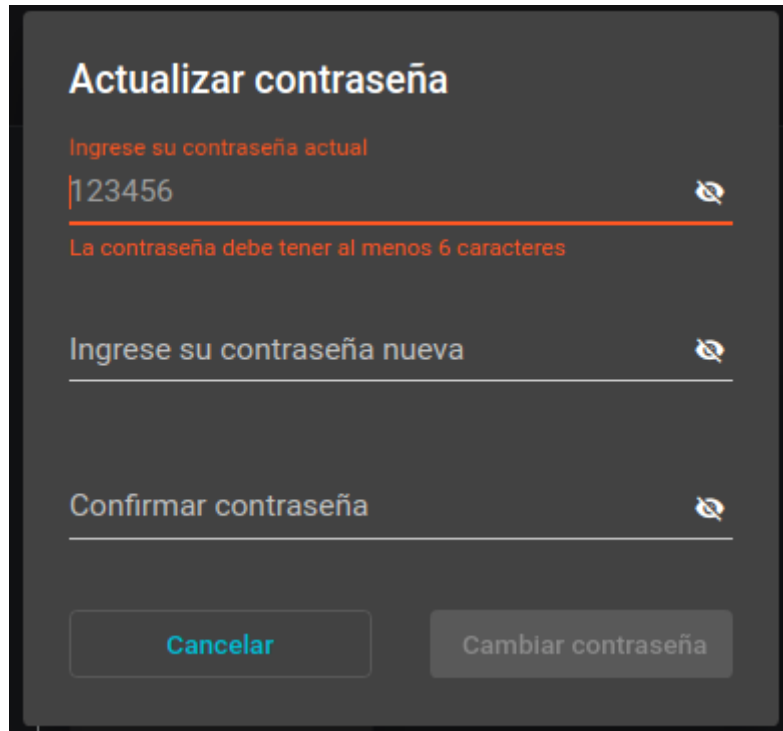


Figura 28: Ventana de visualización de árbitros de un evento.

### Cambio de contraseña y de email

En caso de que el usuario se encuentre con la sesión iniciada y quiera cambiar su contraseña, simplemente ingresa a su perfil y apretando el botón “cambiar contraseña” se le solicitarán los siguientes datos:

- Contraseña actual.
- Nueva contraseña.
- Confirmar nueva contraseña.



The screenshot shows a dark-themed form titled "Actualizar contraseña". It contains three input fields, each with a label and a toggle icon (an eye with a slash) to the right. The first field is labeled "Ingrese su contraseña actual" and contains the text "123456". Below this field is a red error message: "La contraseña debe tener al menos 6 caracteres". The second field is labeled "Ingrese su contraseña nueva" and is empty. The third field is labeled "Confirmar contraseña" and is empty. At the bottom of the form, there are two buttons: "Cancelar" (in cyan) and "Cambiar contraseña" (in grey).

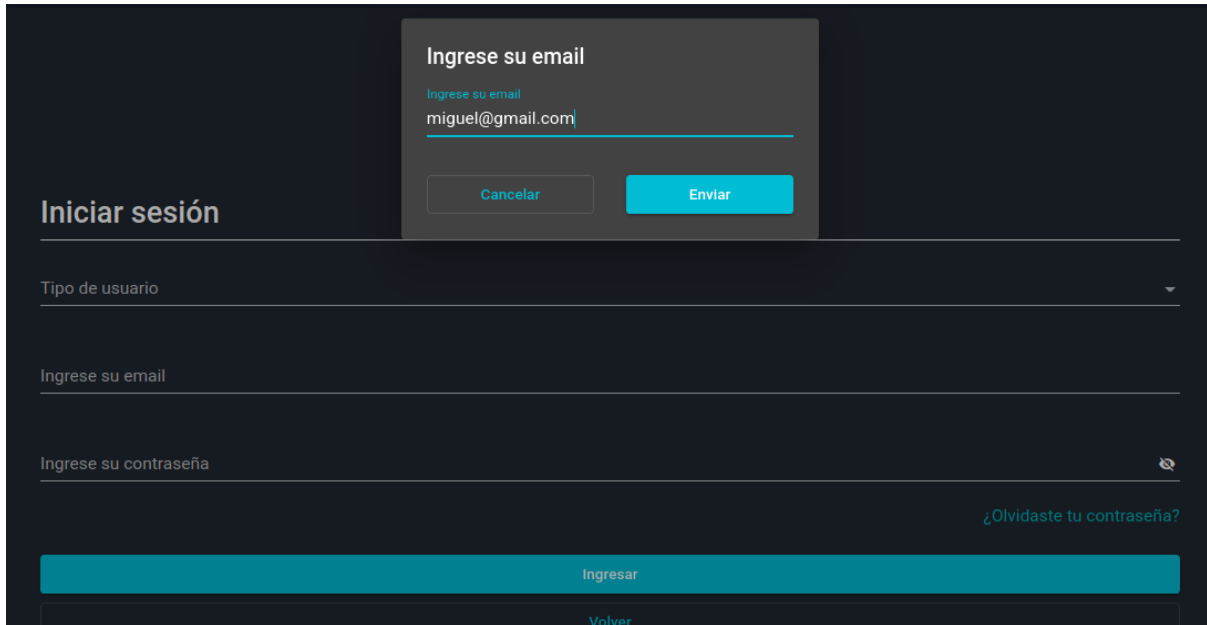
Figura 29: Pantalla que nos muestra cómo cambiar la contraseña.

Una vez realizado esto el sistema procederá a enviar un email para confirmar el cambio de contraseña.



Figura 30: Pantalla de confirmación de contraseña.

En caso de que el usuario no se acuerde de su contraseña y no esté logueado, podrá hacer clic en el enlace de ¿Olvidaste tu contraseña? Ubicado en la pantalla de iniciar sesión.



Ingresar su email

Ingresar su email  
miguel@gmail.com

Cancelar Enviar

Iniciar sesión

Tipo de usuario

Ingresar su email

Ingresar su contraseña

¿Olvidaste tu contraseña?

Ingresar

Volver

Figura 31: Pantalla de recuperación de contraseña.

El sistema enviará un email como el siguiente:

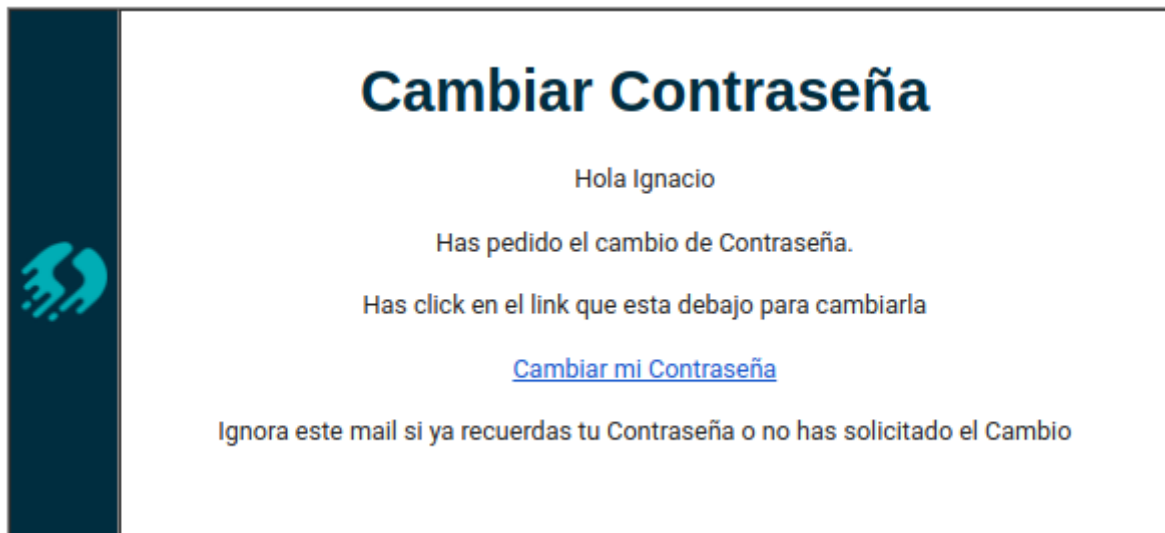


Figura 32: Pantalla del email enviado para cambiar la contraseña.

Al hacer clic en el enlace se le mostrará una pantalla donde puede ingresar su nueva contraseña junto con su confirmación.




Figura 33: Pantalla de solicitud de cambio de contraseña.

El procedimiento para poder actualizar su email es muy parecido, pero con la diferencia que ahora se le solicita la contraseña actual y el nuevo email.

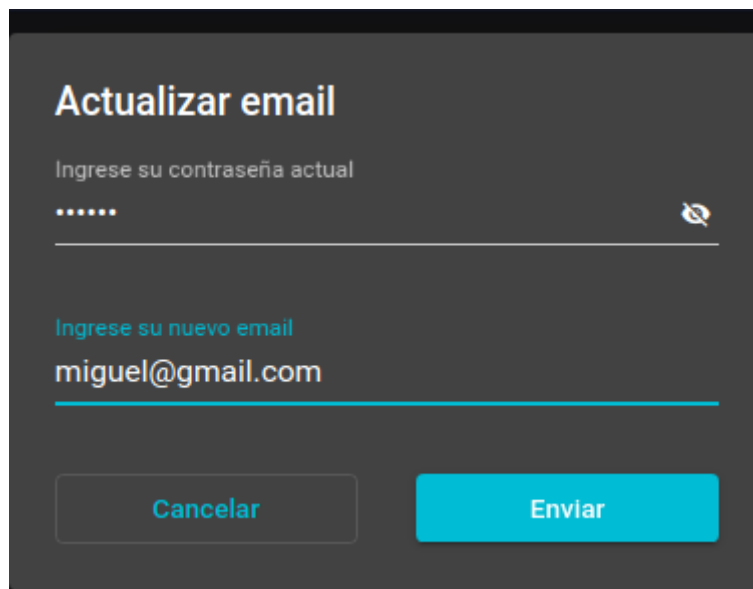


Figura 34: Pantalla de cambio de email.

Una vez que se apretó el botón enviar, el sistema le envía un email al nuevo correo ingresado para confirmar el cambio.



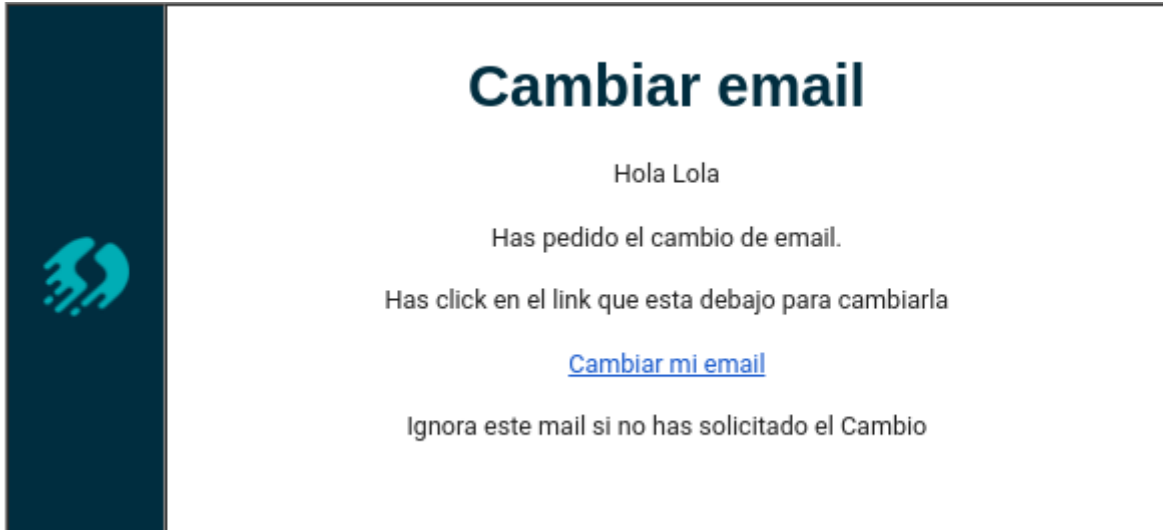


Figura 35: Pantalla de confirmación de cambio de email

Realizar feedback

Cualquier tipo de usuario básico puede realizar un feedback sobre un usuario club, o sobre un evento. Para esto, el usuario debe ubicarse en la pantalla de la categoría del evento a realizar feedback, y clickear el botón “Feedback”.

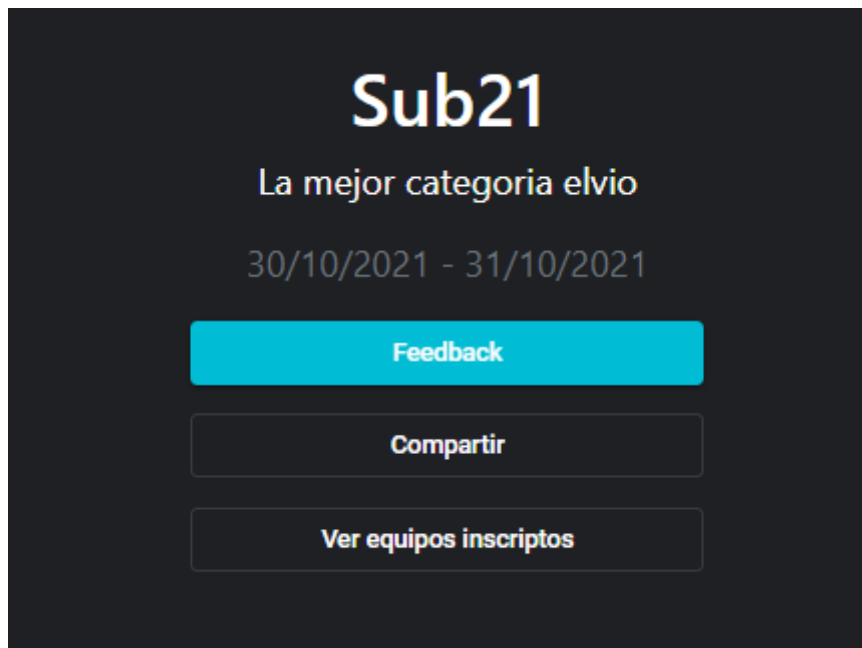


Figura 36: Botón de feedback en categoría de evento

Al hacer clic en el botón inscribirse, aparecerá una ventana con un formulario donde tendrá que llenar los datos del feedback para el evento y del club.

Contanos que te parecio el evento...  
el evento estuvo boenisimo, me gustaria dejar de ser tan bueno

★ ★ ★ ★ ★

### Feedback del club: River Plate

Contanos que te parecio el club...  
el mejor club ever

★ ★ ★ ★ ★

Cancelar Enviar

Figura 37: Ventana de feedback

Darse de baja del sistema

Un usuario puede darse de baja en cualquier momento, para lo cual lo que tendrá que hacer es, dirigirse a su perfil con el botón con forma de persona y hacer clic en configuración.

Esto lo redirigirá a la pantalla de configuración de su cuenta, donde finalmente hará clic en el botón "Darse de baja del sistema". Luego, se le solicitará su contraseña para confirmar que realmente quiere darse de baja, como muestra la siguiente imagen:

¿Está seguro que desea darse de baja en el sistema?

Ingrese su contraseña actual

.....

Cancelar Enviar

Figura 38: Confirmación para darse de baja del sistema

Una vez que el usuario se encuentra dado de baja ya no podrá ingresar al sistema, y si intenta iniciar sesión se mostrará un mensaje de error informando que dicho usuario se encuentra dado de baja.

#### Modificar redes sociales

El usuario puede clicar en el botón “vincular red social” en la configuración de su perfil para modificar los datos de sus redes sociales por las que quiere que los demás usuarios lo contacten.



El formulario muestra un título "Ingrese sus redes sociales" y tres campos de texto con sus respectivos títulos: "Ingrese su link al perfil de twitter", "Ingrese su link al perfil de Instagram" (con el ejemplo <https://www.instagram.com/clubgodoycruzofici>) y "Ingrese su link al perfil de Facebook". Un cuarto campo está etiquetado "Ingrese su link al perfil de LinkedIn" con el ejemplo <https://www.linkedin.com/in/ignacio-starkloff-07>. Al final del formulario hay dos botones: "Cancelar" y "Enviar".

Figura 39: Confirmación para darse de baja del sistema

#### Agregar o quitar rol a usuario

Para agregar o quitar roles a un usuario, este puede acceder a esta opción clicando en el botón “agregar nuevo rol a usuario” o “eliminar rol a usuario”.

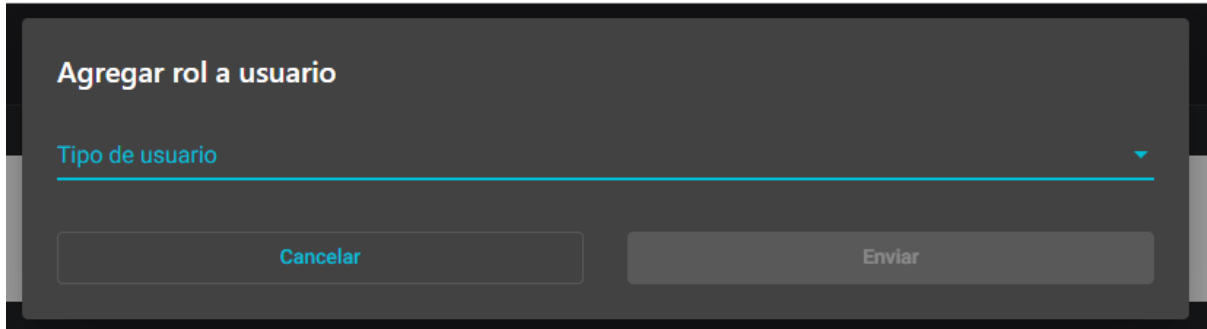


Figura 40: Agregar rol a usuario

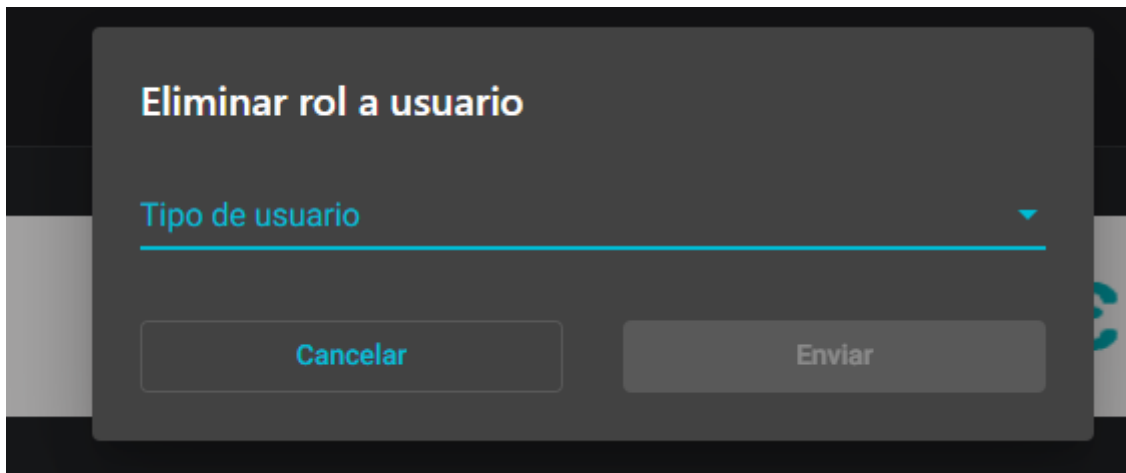


Figura 41: Quitar rol a usuario

### Menús laterales

Cada usuario posee en su menú lateral (sidebar), distintos apartados con las acciones que pueden realizar, dependiendo del tipo de usuario con el que inicio sesión en el sistema. A continuación se detallan las funcionalidades para cada tipo de usuario:

## Sidebar administrador

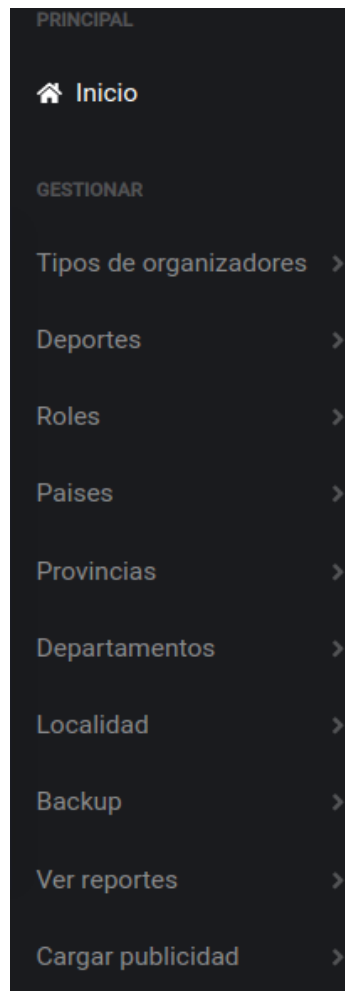


Figura 42: Sidebar del administrador

### Cargar publicidades

El usuario administrador puede en cualquier momento, cargar la publicidad que será mostrada en la aplicación. Para realizar esto, deberá estar haber iniciado sesión como administrador y hacer clic en “cargar publicidad”. Se le cargará un formulario donde podrá determinar las características de la publicidad, como por ejemplo, la imagen y el lugar donde se mostrará la misma y la URL a donde será redirigido el usuario que la presione. Las siguientes imágenes nos muestran como un administrador puede cargar la publicidad en el sistema.

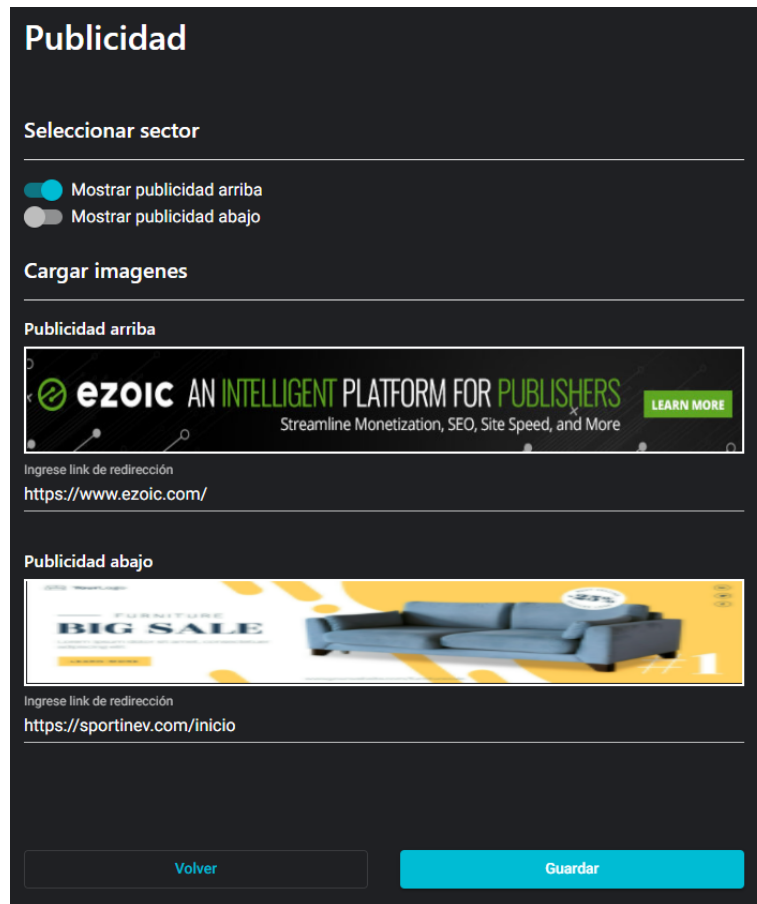


Figura 43: Pantalla de carga de publicidad en la aplicación

Una vez cargada la publicidad se podrá observar la publicidad en toda la aplicación:

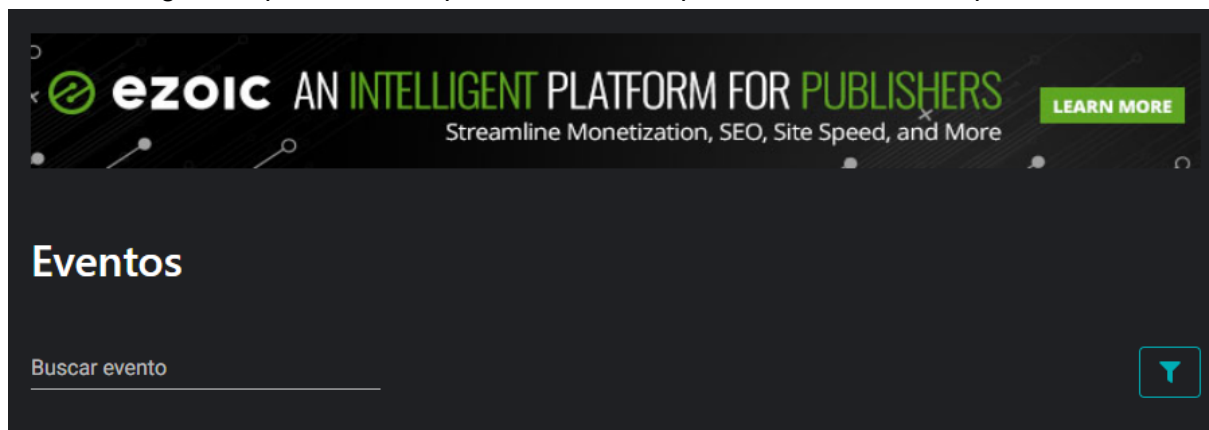



Figura 44: Pantalla que muestra la publicidad a lo largo del sistema

### Roles especiales

Dentro del sidebar del administrador podremos ver varios apartados relacionados con la gestión de roles, como pueden ser: ver rol, crear rol, ver usuarios con roles especiales y asignar roles especiales a los usuarios.

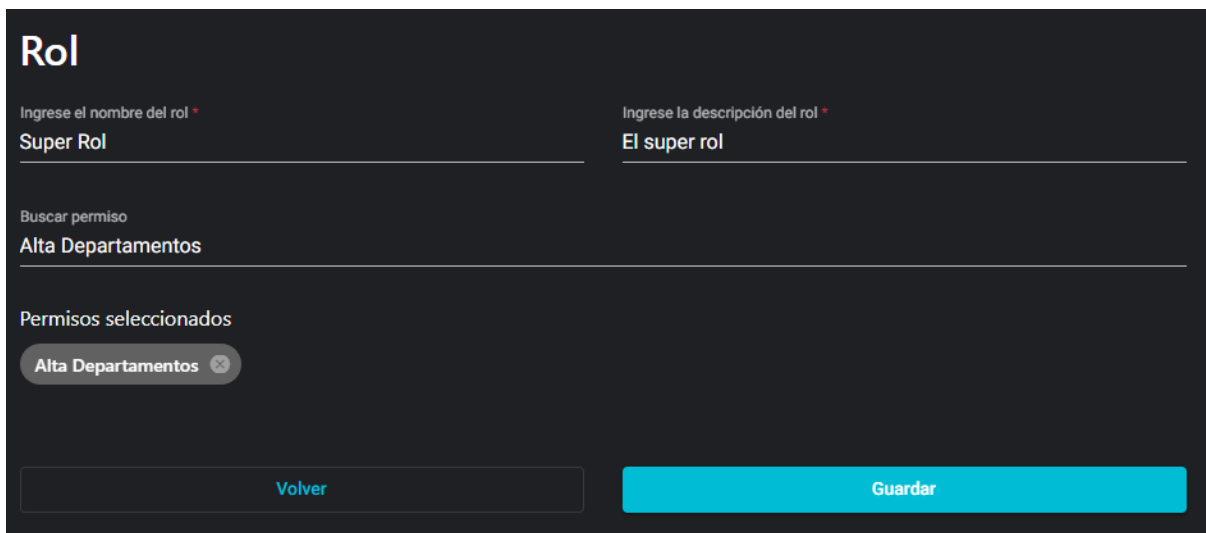


Nombre	Descripción	Permisos	Actions
Miron	Un rol que solo puede mirar y no puede ejecutar ninguna accion		
Probanding	Una descripcion para cambiar la descripcion		
Profesor de gimnasia	Rol para que cargue los deportes		
Supervisor de deportes	Rol encargado de crear, modificar y dar de baja deportes		

Items por página: 10 1 - 10 of 10

Figura 45: Visualización de los roles en el sistema

También tenemos un apartado para poder crear y modificar un rol, donde se ingresa el nombre del rol, las descripciones y los distintos permisos, como se muestra en las imágenes a continuación:



**Rol**

Ingrese el nombre del rol \*  
 Super Rol

Ingrese la descripción del rol \*  
 El super rol

Buscar permiso  
 Alta Departamentos

Permisos seleccionados  
 Alta Departamentos

Volver Guardar

Figura 46: Creación de roles en el sistema

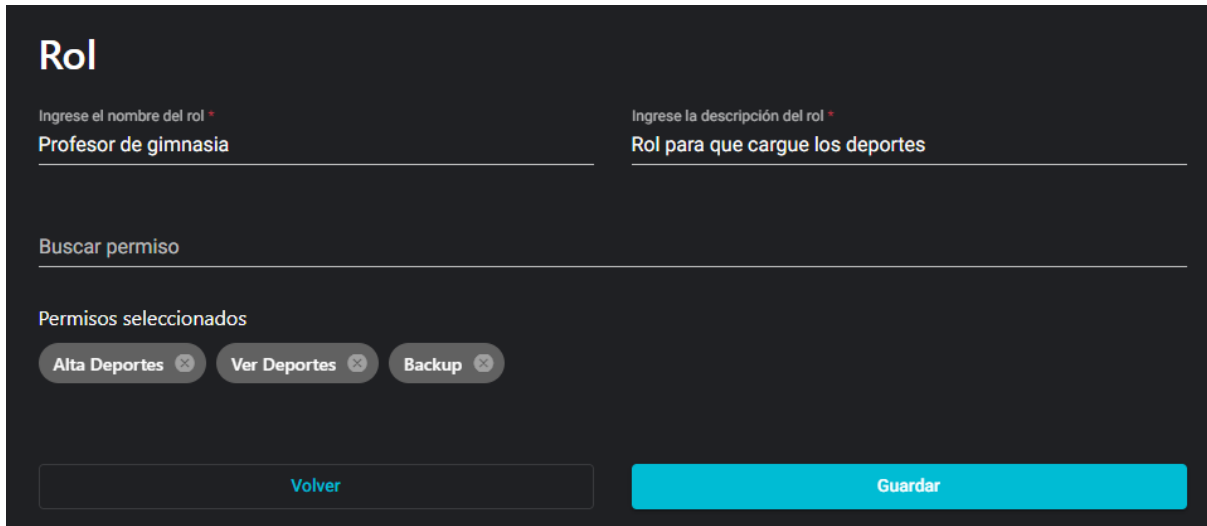
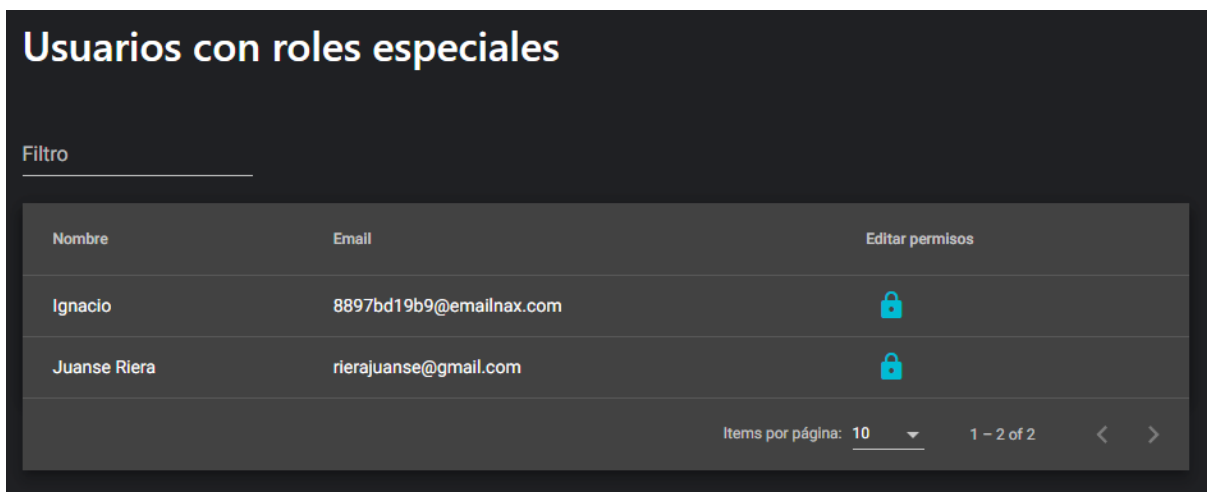




Figura 47: Modificación de un rol en el sistema

Adicionalmente en el mismo apartado podemos ver los usuarios que tienen asignados los roles especiales, como muestra la siguiente imagen:



Nombre	Email	Editar permisos
Ignacio	8897bd19b9@emailnax.com	
Juanse Riera	rierajuanse@gmail.com	

Items por página: 10 1 - 2 of 2

Figura 48: Visualización de los usuarios con roles especiales

La última acción que puede realizar el administrador en este apartado es la de asignar un rol especial a algún usuario en particular. Para efectuar esto deberá seleccionar un rol especial creado anteriormente y asignárselo a un usuario existente en el sistema.



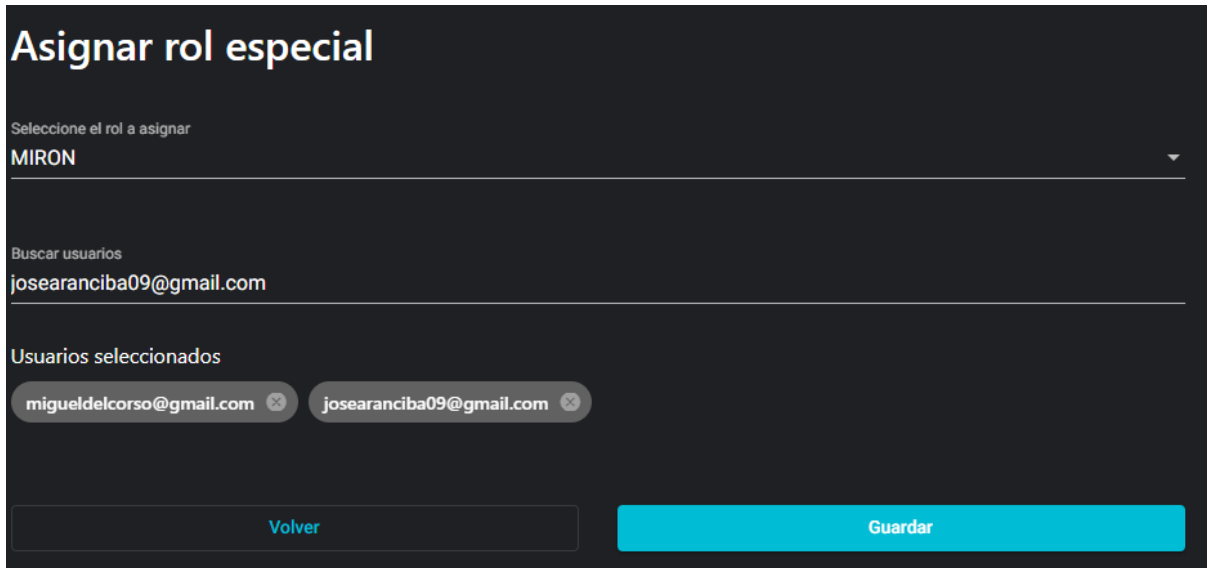


Figura 49: Asignación de roles especiales por parte del administrador

## Backups

El usuario administrador puede en cualquier momento realizar un backup de toda la base de datos del sistema. Para efectuar esto deberá iniciar sesión como administrador y en el sidebar hacer clic en el apartado llamado backup, dentro de ese apartado deberá seleccionar el sub apartado llamado “ver backups”.

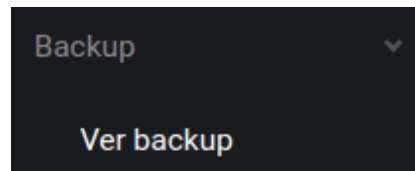


Figura 50: Apartado de backup en el sidebar del administrador

Finalmente en la pantalla nos encontraremos con dos botones:

El botón con el icono de descarga para hacer un backup de la versión actual del sistema.

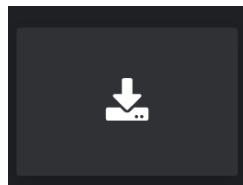


Figura 51: Botón para realizar un backup

El botón con el icono de subida para restaurar una versión anterior del sistema.

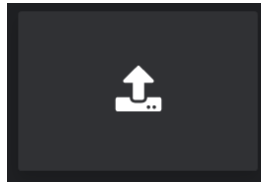


Figura 52: Botón para restaurar un backup

### *Realización del backup*

En caso de que el usuario haga clic en el primer botón, aparecerá un mensaje indicando que el backup se encuentra en proceso de realización:

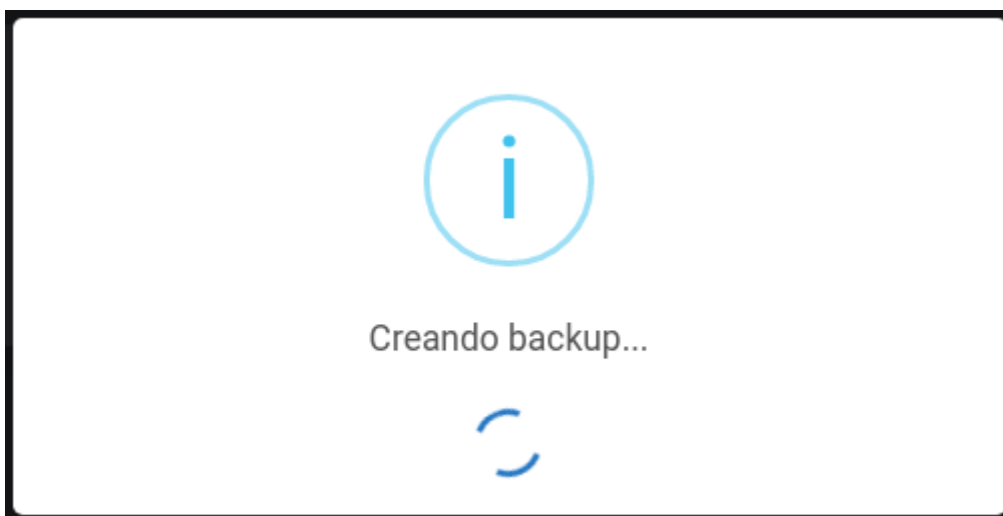


Figura 53: Mensaje de realización de backup en curso

Generalmente, el proceso de realización de backup tarda unos pocos segundos.

Una vez realizado el backup se le mostrará al usuario administrador un mensaje indicándole la correcta realización del backup:

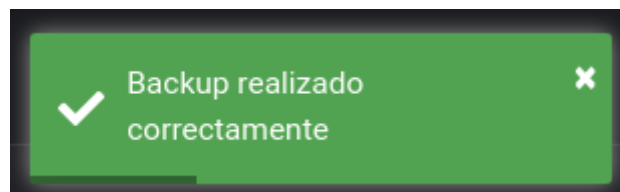


Figura 54: Mensaje de backup realizado correctamente

Adicionalmente en caso de que lo hayamos configurado en el navegador nos saldrá una ventana, en el cual deberemos indicar donde queremos guardar el backup generado.

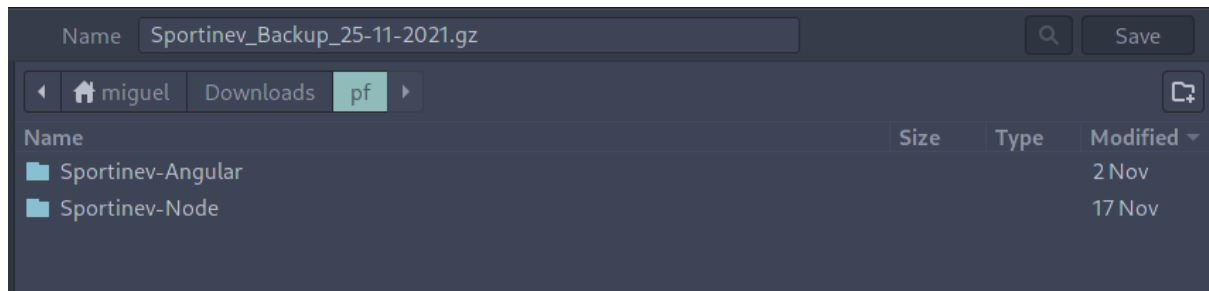


Figura 55: Imagen que nos muestra la ubicación de guardado del backup

Como podemos observar el nombre del backup tiene un formato especial el cual es el siguiente:

`Sportinev_Backup_dd_mm_yyyy.gz`

Donde:

- dd: día del mes actual
- mm: mes del año actual
- yyyy: año actual

Se eligió dicho formato para el nombre, ya que le puede ser de especial utilidad al usuario administrador para una mejor organización y control.

El backup es comprimido por el sistema en un formato .gz, se eligió el mencionado formato para ahorrar espacio, especialmente cuando tenemos cientos o miles de backups guardados.

Una vez que el usuario haya elegido el destino adecuado para el backup, hará clic en el botón guardar y en la parte inferior del navegador podremos observar que efectivamente el backup se descargó.

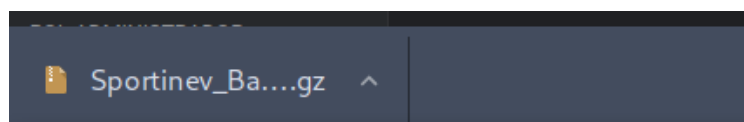


Figura 56: Imagen que nos muestra el backup guardado correctamente

### *Restauración del backup*

En el caso de apretar el segundo botón el administrador seleccionará un backup previo y el sistema procederá a restaurar el sistema al estado en el que se encontraba el sistema al momento en que se realizó el backup.

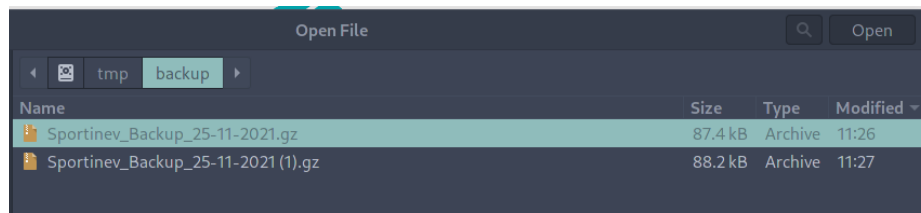


Figura 57: Selección del backup.

El administrador deberá ingresar un backup válido, ya que en caso de querer ingresar cualquier otro tipo de archivo el sistema mostrara el siguiente mensaje:

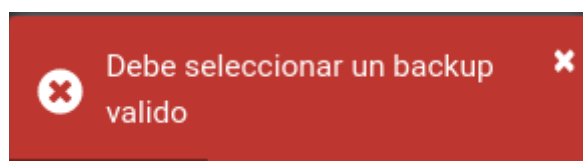


Figura 58: Mensaje de error a la hora de seleccionar el backup.

Una vez que el usuario seleccione un backup correcto y haga clic en el botón de cargar, el sistema procederá a restaurar la base de datos. Mientras se esté restaurando el backup se mostrará el siguiente mensaje:

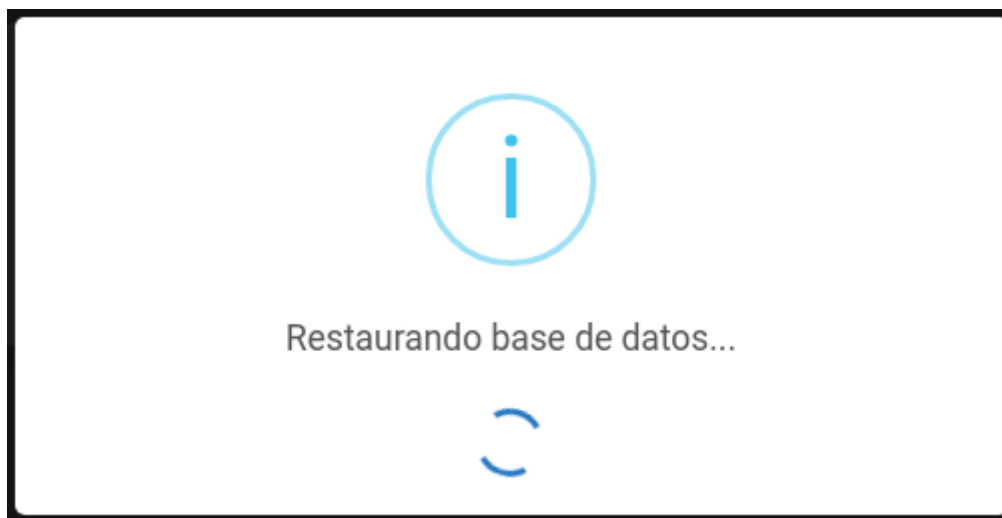


Figura 59: Mensaje de procesamiento del backup

Dicha restauración puede llevar un tiempo dependiendo del tamaño que tenía el sistema cuando se efectuó el backup.

Finalmente, una vez que el backup se realizó correctamente el sistema mostrara el siguiente mensaje:

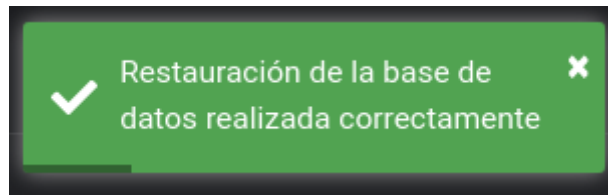


Figura 60: Mensaje de éxito de restauración del backup.

En la siguiente imagen se puede observar la página del backup.

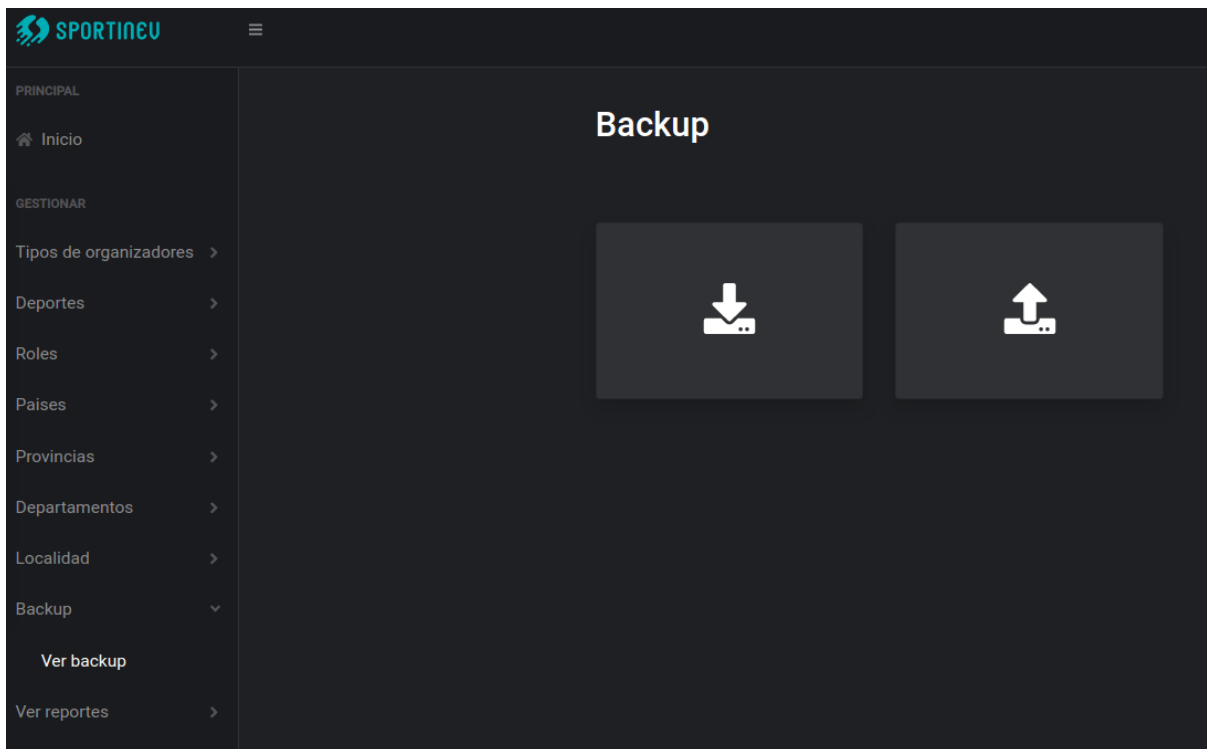


Figura 61: Funcionamiento del backup

Listado de los tipos de organizadores

El usuario con permisos de administrador puede en cualquier momento, visualizar los tipos de organizadores existentes en el sistema, modificarlos e incluso agregar nuevos según le resulte más conveniente.



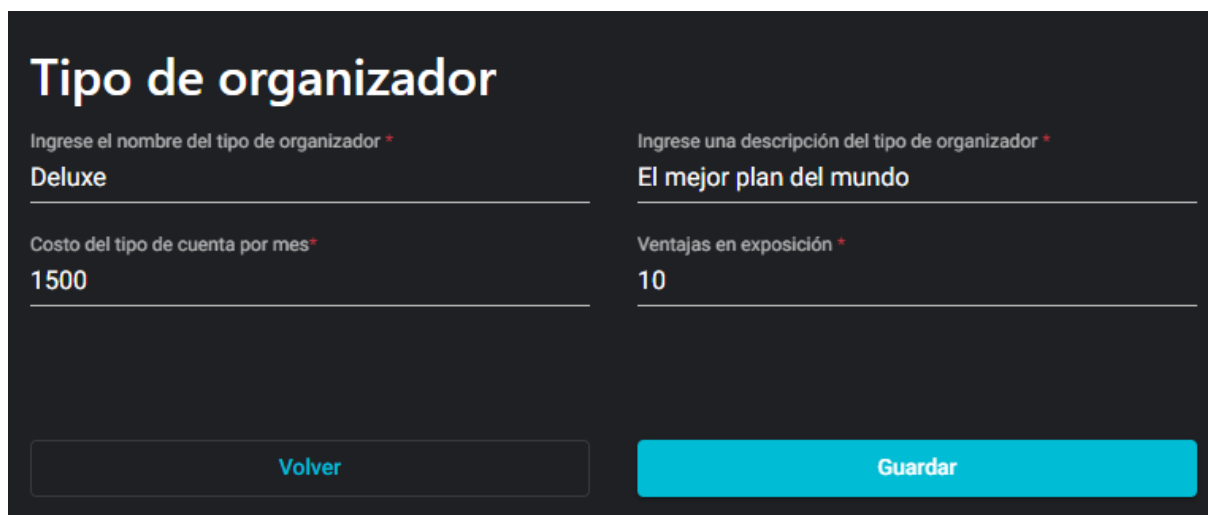
Nombre	Descripción	Precio por mes	Valor para recomendación	Actions
Barato	Pack económico	1	1	 
Gratis	La categoría para la gente que no quiere poner plata en nuestra aplicación :(	0	0.5	 
Premium	La mejor categoría para los organizadores	250	2.97	 

Items por página: 10 1 - 3 of 3

Figura 62: Listado de los tipos de organizadores en el sistema

#### Creación y modificación de los tipos de organizadores

Para crear un nuevo tipo de organizador, el usuario deberá iniciar sesión como administrador y hacer clic sobre el botón “Crear tipo organizador” en el sidebar de la aplicación. Se le mostrará un formulario con distintos datos que deberá llenar y al presionar el botón “Guardar” se generará un nuevo tipo de organizador.



**Tipo de organizador**

Ingrese el nombre del tipo de organizador \*

**Deluxe**

Ingrese una descripción del tipo de organizador \*

**El mejor plan del mundo**

Costo del tipo de cuenta por mes\*

**1500**

Ventajas en exposición \*

**10**

**Volver** **Guardar**

Figura 63: Creación de un tipo organizador en el sistema

Para editar un rol existente, el usuario hará clic en el icono del lapiz, dentro del apartado ver tipos de organizadores y podrá editar los datos del tipo de organizador existente.

## Tipo de organizador

Ingrese el nombre del tipo de organizador \*

**Barato**

---

Costo del tipo de cuenta por mes\*

**1**

---

Ingrese una descripción del tipo de organizador \*

**Pack económico**

---

Ventajas en exposición \*

**1**

---

Volver
Guardar

Figura 64: Modificación de un tipo organizador en el sistema

### Visualización de los deportes

El usuario administrador tiene un apartado para poder visualizar los deportes que se encuentran activos en el sistema. Le muestra información útil como el nombre, descripción, cantidad de equipos mínimos y máximos, etc.

## Deportes

Agregar

Filtro

Nombre	Descripción	Cantidad de equipos mínimo por cancha	Cantidad de equipos máximos por cancha	Cantidad integrantes	Tipos de eventos	Actions
Boxeo	dale veni toma pum	2	2	1		
Futbol	El fulbito	2	2	11		
Futbol-Tenis	El Futbol-Tenis	2	2	1,2,3,4,5		
Hockey	futbol pero con palos y gente que no se queja	22	23	11		
Tenis	El Tenis	2	2	1,2,3,4,5		

Items por página: 10
1 - 5 of 5
< >

Figura 65: Pantalla de visualización de los deportes

Por cada uno de los deportes se puede realizar acciones, como editar el deporte o eliminarlo.

Visualización de países, provincias, departamentos y localidades

El administrador en el sidebar de la izquierda puede ver los botones “Ver países”, “Ver provincias”, “Ver departamentos” y “Ver localidades”, que le permitirá dirigirse a las páginas para ver países, provincias, departamentos y localidades respectivamente. En las mismas se encontrará con una tabla con todos los registros creados y la posibilidad de editarlo o eliminarlos.







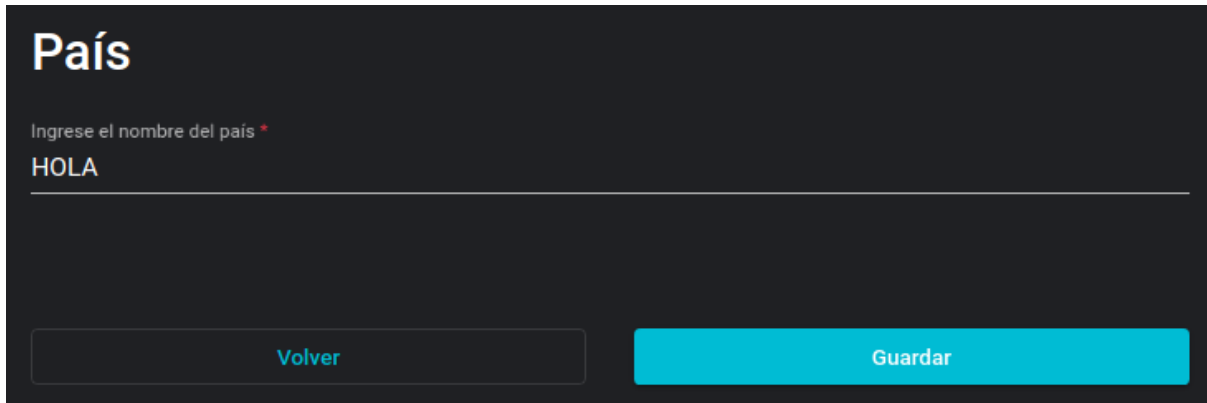
Nombre	País al que pertenece	Actions
Buenos Aires	Argentina	 
Catamarca	Argentina	 
Chaco	Argentina	 
Chubut	Argentina	 
Corrientes	Argentina	 
Córdoba	Argentina	 
Entre Ríos	Argentina	 
Formosa	Argentina	 
Juanse	Argentina	 
Jujuy	Argentina	 

Figura 66: Pantalla de visualización de las provincias

Creación de países, provincias, departamentos y localidades.

Para poder crear un nuevo país, el usuario deberá iniciar sesión como administrador y presionar el botón “Crear país” que se encuentra en el sidebar de la izquierda. Luego se le mostrará un formulario como el que se muestra a continuación que deberá completar con información válida.





**País**

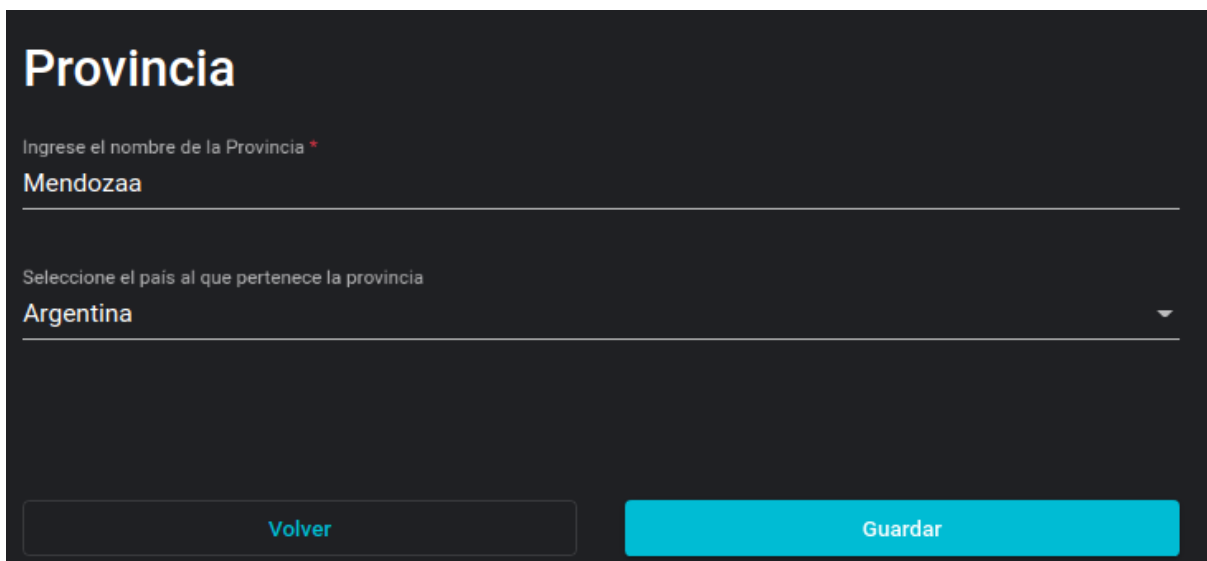
Ingrese el nombre del país \*

HOLA

Volver Guardar

Figura 67: Creación del país

Para poder crear una nueva provincia, el usuario deberá iniciar sesión como administrador y presionar el botón “Crear provincia” que se encuentra en el sidebar de la izquierda. Luego se le mostrará un formulario como el que se muestra a continuación que deberá completar con información válida.



**Provincia**

Ingrese el nombre de la Provincia \*

Mendozaa

Seleccione el país al que pertenece la provincia

Argentina

Volver Guardar

Figura 68: Creación de la provincia

Para poder crear un nuevo departamento, el usuario deberá iniciar sesión como administrador y presionar el botón “Crear departamento” que se encuentra en el sidebar de la izquierda. Luego se le mostrará un formulario como el que se muestra a continuación que deberá completar con información válida.

**Departamento**

Ingrese el nombre de el Departamento \*

Godoy Cruz

País: Argentina

Provincia: Mendoza

Volver Guardar

Figura 69: Creación del departamento

Para poder crear un nuevo país, el usuario deberá iniciar sesión como administrador y presionar el botón “Crear país” que se encuentra en el sidebar de la izquierda. Luego se le mostrará un formulario como el que se muestra a continuación que deberá completar con información válida.

**Localidad**

Ingrese el nombre de la Localidad \*

San Jose

País: Argentina

Provincia: Córdoba

Departamento al que pertenece la localidad: Calamuchita

Volver Guardar

Figura 70: Creación de la localidad

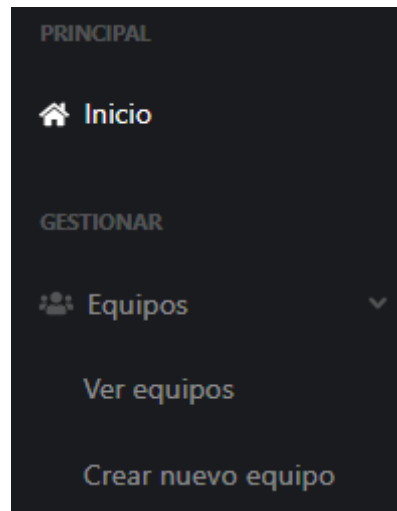


Figura 71: Sidebar del deportista

### Visualización de equipos

El usuario con rol deportista puede visualizar los equipos que posee, en una muy conveniente lista. Cada una de las filas de la tabla contiene información útil como el nombre, la descripción y una lista de los integrantes que forman parte del equipo.

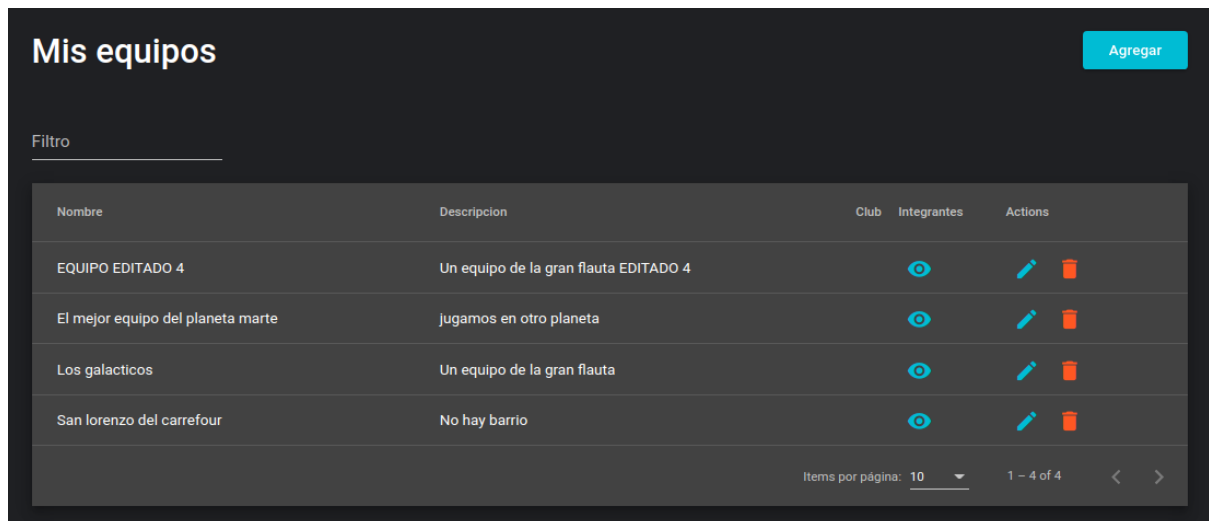


Figura 72: Pantalla de visualización de equipos

### Modificación y alta de equipos

Tanto para crear como para modificar un equipo el usuario deberá completar el mismo formulario. Para esto, el usuario deberá iniciar sesión como deportista e ir a la pantalla de visualización de los equipos. En esta pantalla si lo que desea es crear un nuevo equipo deberá presionar el botón “Agregar”.

Si lo que desea es editar un equipo existente, deberá presionar el botón con el icono de lápiz, que se encuentra en la fila del equipo que quiere editar.

En cualquier caso, será le mostrará un formulario como el que se muestra a continuación, que deberá llenarlo con datos correctos.

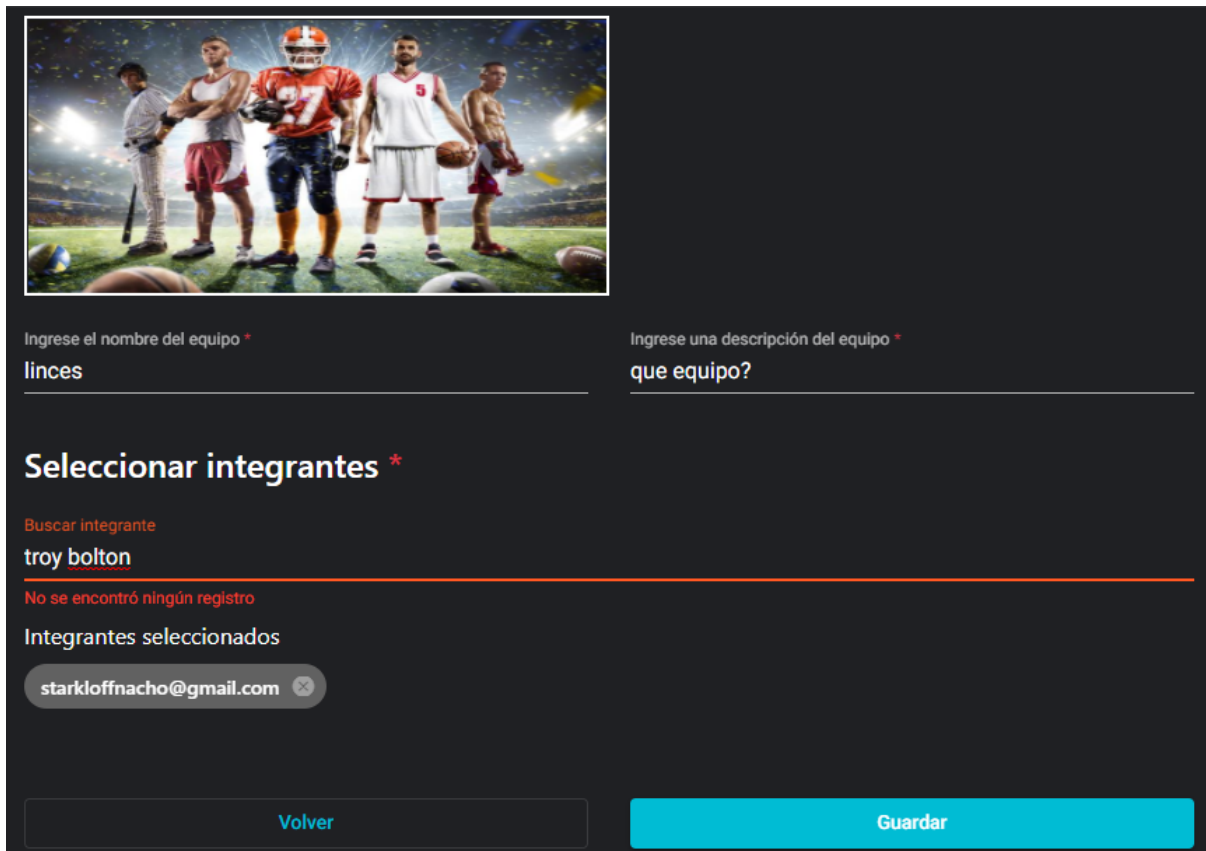


Figura 73: Pantalla de alta y modificación de equipos

Una vez cargados los datos, el usuario debe clicar en guardar y ya podrá ver los cambios en la pantalla de visualización de equipos.

### Inscripción a eventos

En la pantalla principal del sistema se pueden observar todos los eventos que se han generado. Para poder inscribirse en uno de ellos, el usuario deberá haber iniciado sesión como deportista y deberá presionar el botón “Ver más” del evento al que se desea inscribir.

Podrá ver la información más importante del evento, incluyendo las categorías del mismo. Si decide inscribirse deberá presionar el botón “Ver más” de la categoría en la que quiere participar.

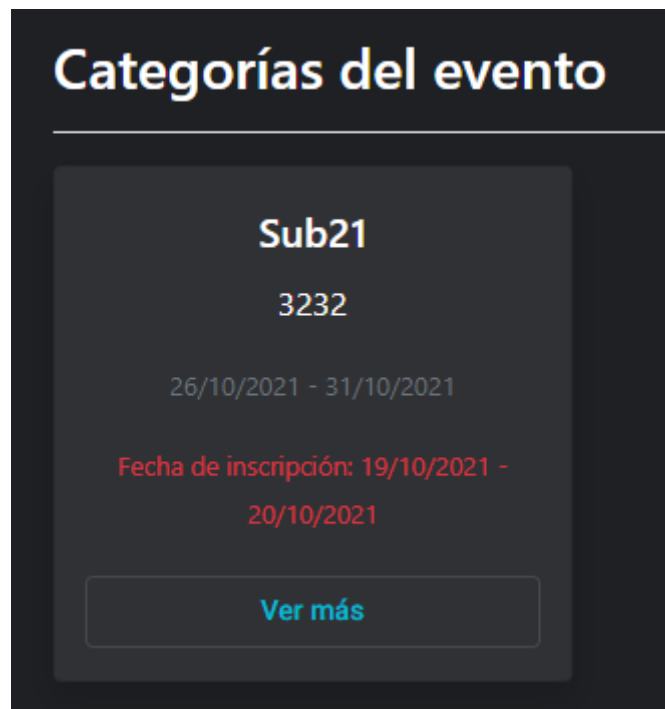


Figura 74: Pantalla de eventos, sección categorías de un evento

Al presionarlo se muestra una pantalla con toda la información de la categoría y un botón para inscribirse.



Figura 75: Pantalla de categoría de un evento

Si el usuario cliquee en inscribirse, se le mostrará un formulario que deberá completar con los horarios de preferencia del deportista, observaciones, el medio de pago con el que desea pagar la inscripción y finalmente el equipo con el cual se inscribirá al evento.

Una vez que el deportista ha llenado el formulario debe seleccionar el botón enviar para finalizar la inscripción.

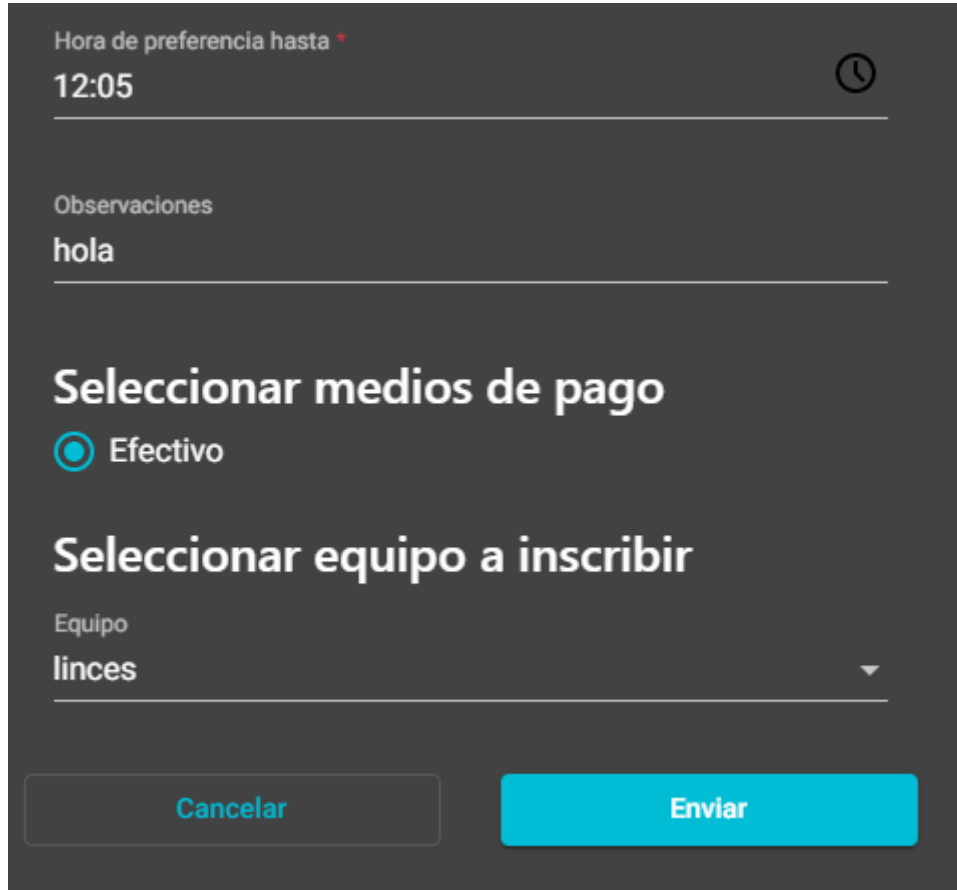


Figura 76: Pantalla de inscripción a categoría de un evento

#### Crear deporte favorito

El deportista puede seleccionar sus deportes favoritos en el menú de configuración del perfil. El usuario deberá presionar el botón “Deportes favoritos”. El sistema le mostrará una pantalla donde deberá introducir sus deportes favoritos y presionar “Guardar cambios”.

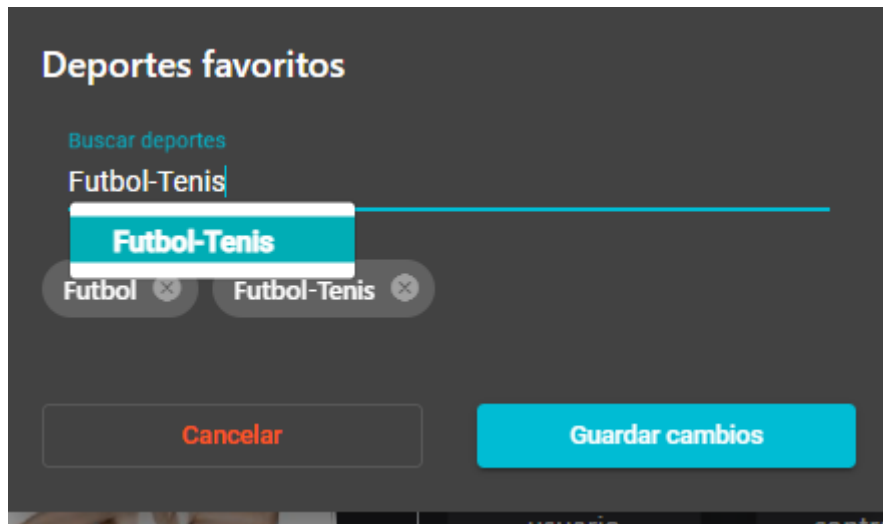


Figura 77: Creación de deporte favorito

### Sidebar organizador

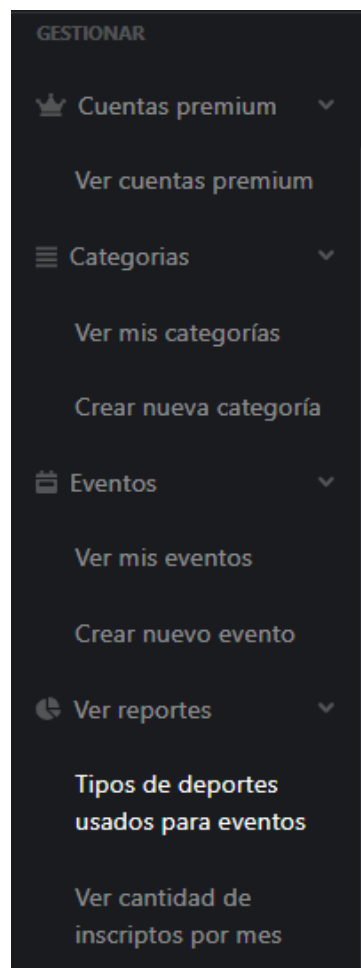
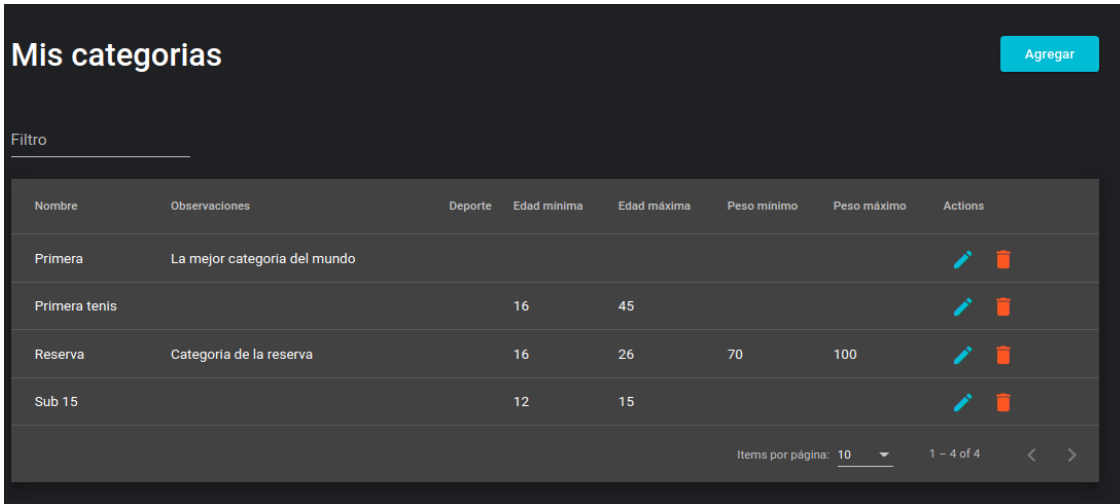










Figura 78: Sidebar del organizador

### Visualización de las categorías

El organizador puede consultar la información de las categorías generadas, como puede ser el nombre, las observaciones, el deporte, la edad mínima y la máxima. Además de contar con distintas acciones como editar y borrar.



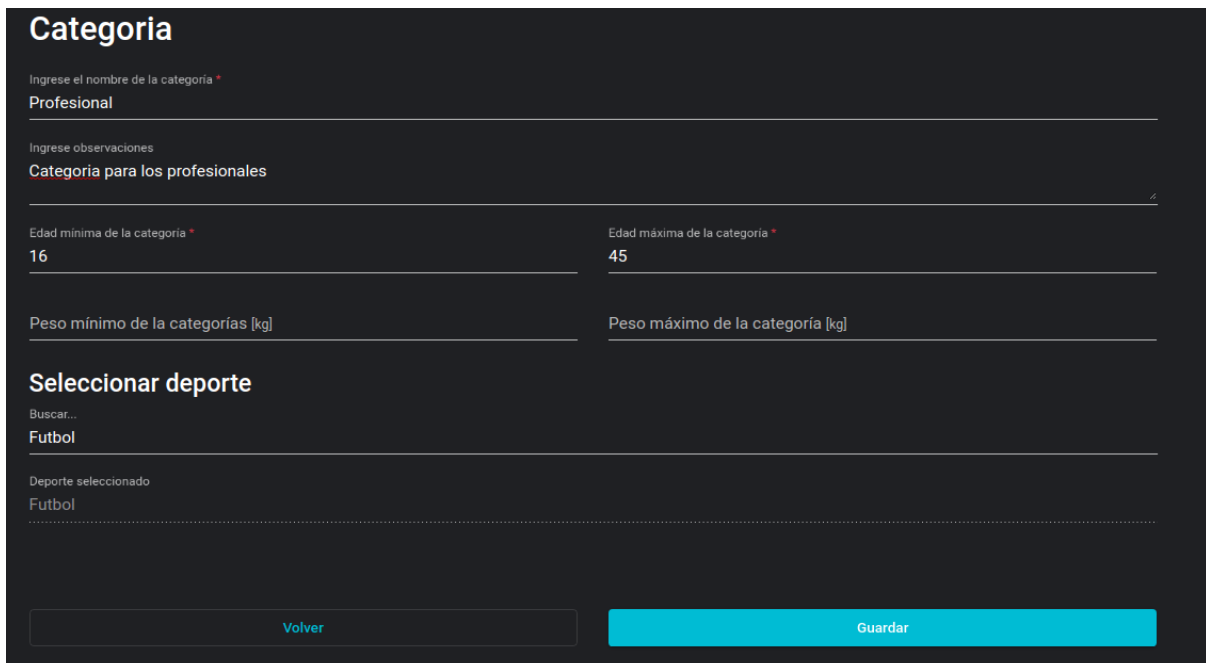
Nombre	Observaciones	Deporte	Edad mínima	Edad máxima	Peso mínimo	Peso máximo	Actions
Primera	La mejor categoría del mundo						 
Primera tenis			16	45			 
Reserva	Categoría de la reserva		16	26	70	100	 
Sub 15			12	15			 

Items por página: 10 1 - 4 of 4

Figura 79: Pantalla de visualización de las categorías.

### Modificación y alta de las categorías

En caso de que el usuario organizador desee crear una categoría o modificar una existente deberá completar el mismo formulario. En el primer caso, deberá presionar el botón “Agregar” caso contrario deberá hacer clic en el botón con forma de lápiz en el apartado de las acciones de la categoría que quiere editar. Luego se le mostrará un formulario como el que se muestra a continuación, que deberá completar con datos válidos.



### Categoria

Ingrese el nombre de la categoría \*

Profesional

Ingrese observaciones

Categoría para los profesionales

Edad mínima de la categoría \*

16

Edad máxima de la categoría \*

45

Peso mínimo de la categorías [kg]

Peso máximo de la categoría [kg]

#### Seleccionar deporte

Buscar...

Futbol

Deporte seleccionado

Futbol

Volver Guardar

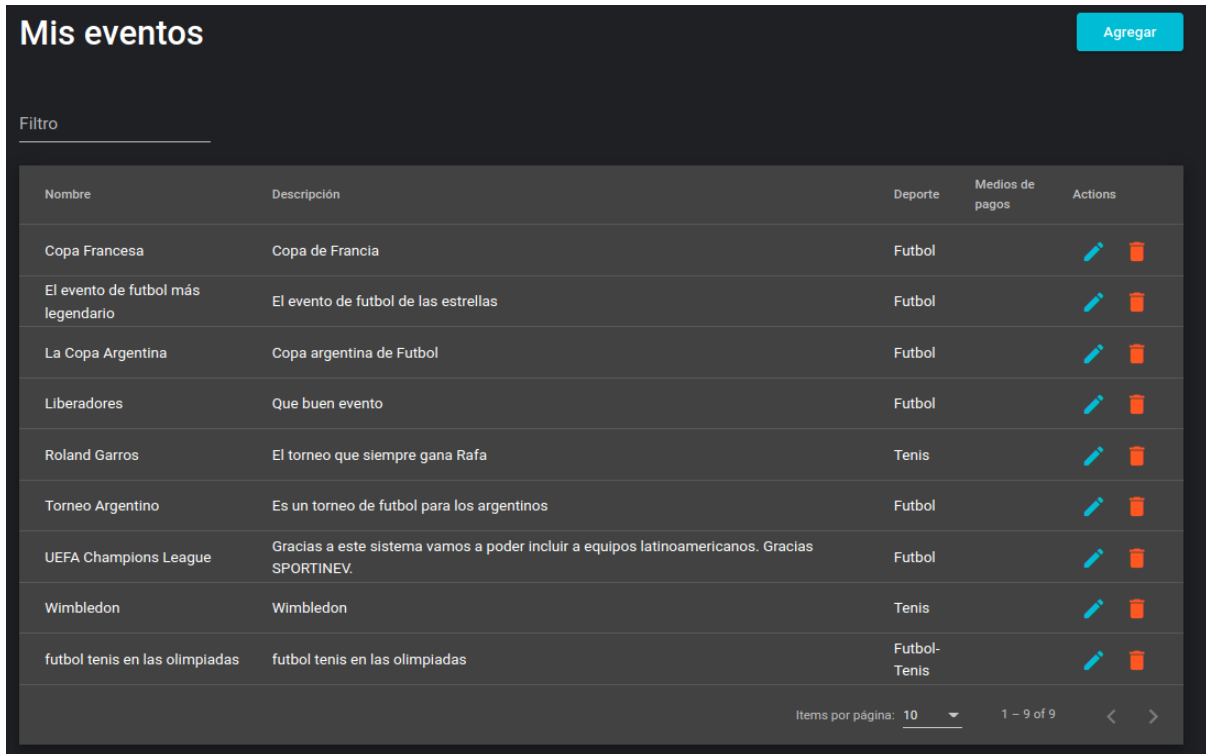




















Figura 80: Pantalla de modificación de las categorías.

Cabe resaltar que los campos que tienen un \* significa que son obligatorios, como por ejemplo el nombre de la categoría.

#### Visualización de eventos de un organizador

En esta pantalla el organizador podrá visualizar los eventos que se encuentran en el sistema, y en caso de que lo desee puede crear un nuevo evento.



Nombre	Descripción	Deporte	Medios de pagos	Actions
Copa Francesa	Copa de Francia	Futbol		 
El evento de futbol más legendario	El evento de futbol de las estrellas	Futbol		 
La Copa Argentina	Copa argentina de Futbol	Futbol		 
Liberadores	Que buen evento	Futbol		 
Roland Garros	El torneo que siempre gana Rafa	Tenis		 
Torneo Argentino	Es un torneo de futbol para los argentinos	Futbol		 
UEFA Champions League	Gracias a este sistema vamos a poder incluir a equipos latinoamericanos. Gracias SPORTINEV.	Futbol		 
Wimbledon	Wimbledon	Tenis		 
futbol tenis en las olimpiadas	futbol tenis en las olimpiadas	Futbol-Tenis		 

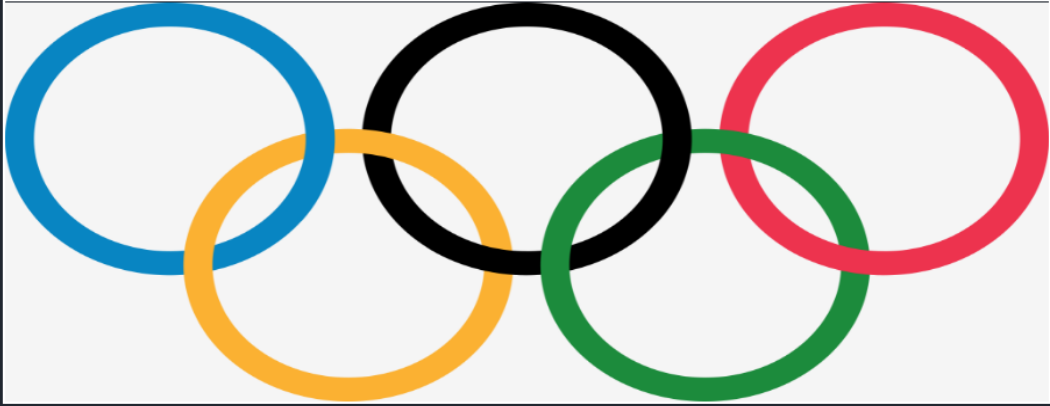
Items por página: 10 1 - 9 of 9

Figura 81: Pantalla que nos muestra todos los eventos del sistema.

#### Creación de eventos

En caso de que el usuario desee crear un nuevo evento, deberá hacer clic en el botón "Agregar" que se puede ver en la pantalla anterior. Una vez realizado esto, se le mostrará un formulario con datos básicos de un evento, que deberá completar con información válida.

## Evento



Ingrese el nombre del evento \*

Super Evento

Ingrese una descripción del evento

Este es un super evento

### Seleccionar tipo de evento

Liga

Copa

### Seleccionar medios de pago

Efectivo

Mercado Pago

### Seleccionar deporte

Buscar...

Futbol-Tenis

Deporte seleccionado

Futbol-Tenis

[Volver](#) [Siguiente](#)

Figura 82: Pantalla que nos muestra el inicio de la creación de un evento

Una vez ingresados esos datos básicos el sistema mostrará las categorías que tiene el evento. Para poder crear una categoría el organizador deberá hacer clic en el botón “Crear categoría”



## Categoría del evento



The screenshot shows a dark-themed interface for event categories. At the top, the title 'Categoría del evento' is displayed in white. Below the title, there are three buttons arranged horizontally. The first button on the left is labeled 'Volver' in orange text. The second button on the right is labeled 'Crear categoría' in light blue text. Below these two buttons, there is a single, wide button labeled 'Crear evento' in white text, which is highlighted with a bright cyan background.

Figura 83: Pantalla para agregar categorías al evento

Se le mostrará un formulario como el que se muestra a continuación, que deberá completar con datos válidos.

### Categoría

Fecha de inicio de inscripción \*   Fecha de finalización de la inscripción \*  

Fecha de inicio del evento para categorí...   Fecha de finalización del evento para ca...  

Descripción\*  Número máximos de equipos \*

Precio de evento \*  Buscar categoría...



Categoría seleccionada


### Agregar clubes

Descripción \*  Buscar club...

Club seleccionado \*

#### Jornada 1

Fecha \*   Hora desde \*  

Hora hasta \*   Observaciones

#### Seleccionar canchas

Primero debe seleccionar el club, la fecha y los horarios. Se mostraran las canchas disponibles segun estos datos.

### Ver clubes

Club	Descripción	Actions
Por favor cargue al menos un club en la categoría		

Figura 84: Pantalla que nos muestra la creación de una categoría del evento

Una vez que se cargó el club apretando en el botón correspondiente, se muestran todos los clubes participantes de esa categoría. Cabe mencionar que es una lista provisoria porque los clubes deben aceptar dicha solicitud.

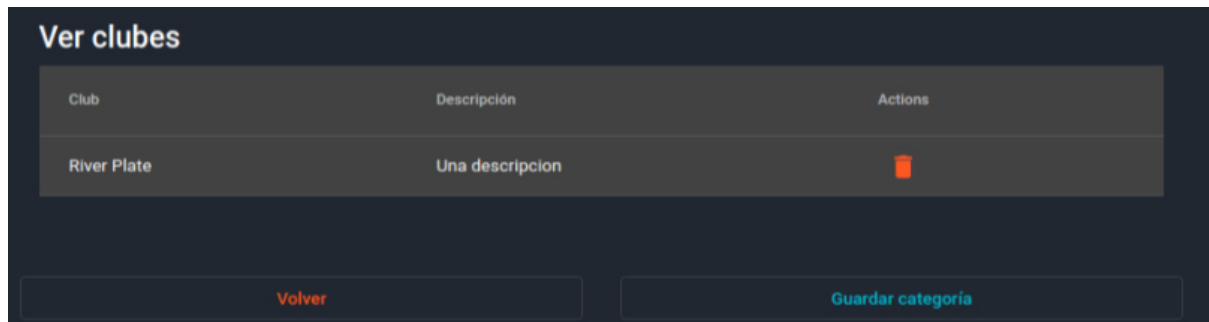


Figura 85: Pantalla de visualización de los clubes de la jornada.

En un evento puede haber muchas categorías, por lo que se puede cargar una nueva categoría realizando otra vez los pasos anteriores.

Finalmente, el organizador deberá presionar el botón “Crear evento” para crear el evento y enviar la solicitud de participación a los clubes que seleccionó.

#### Organizar Eventos

El organizador puede organizar sus eventos al hacer clic en el icono con forma de reloj, en la columna “organizar eventos”, en la tabla de visualización de los eventos. Una vez cliqueado, el sistema redirige al usuario a la página del evento en modo organizador, donde podrá realizar las acciones mostradas en la imagen:

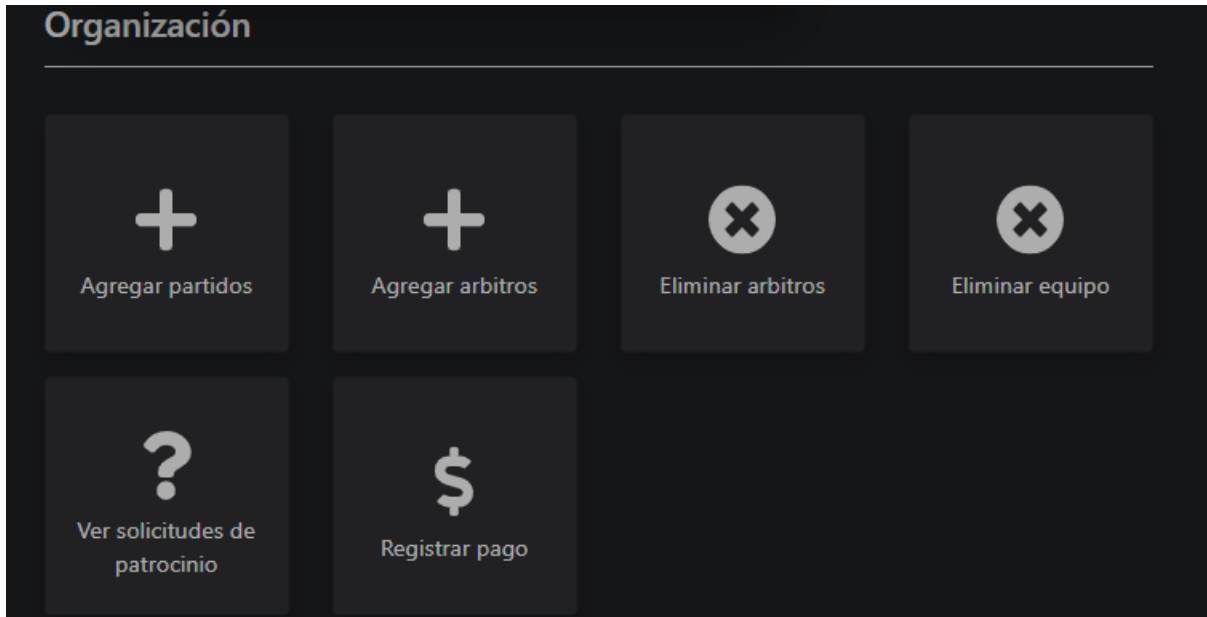
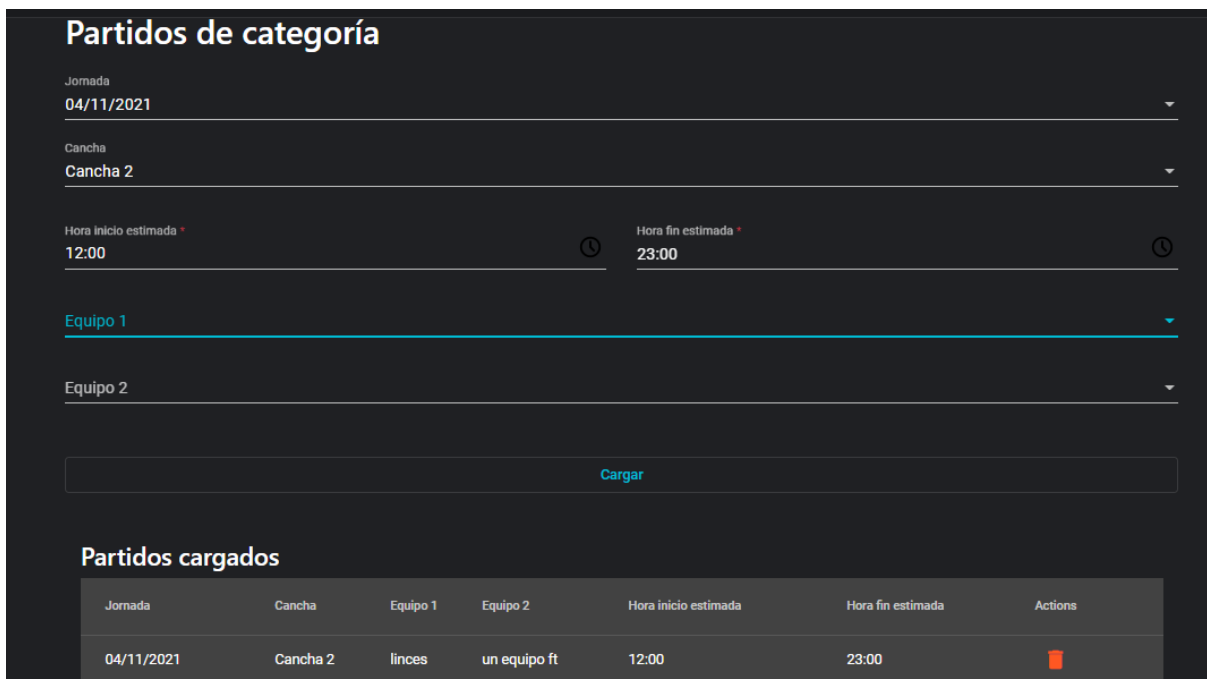


Figura 86: Pantalla de organización de evento

### Agregar partidos

Para agregar un partido, el organizador deberá clicar el botón "Agregar partidos", esto lo redirigirá a la página de carga de partidos, en la que deberá llenar un formulario como el que se muestra a continuación, con datos válidos.



The screenshot shows a form titled "Partidos de categoría" with the following fields and controls:

- Jornada:** 04/11/2021 (dropdown menu)
- Cancha:** Cancha 2 (dropdown menu)
- Hora inicio estimada:** 12:00 (clock icon)
- Hora fin estimada:** 23:00 (clock icon)
- Equipo 1:** (dropdown menu)
- Equipo 2:** (dropdown menu)
- Cargar:** (button)

Below the form is a table titled "Partidos cargados" with the following data:


Jornada	Cancha	Equipo 1	Equipo 2	Hora inicio estimada	Hora fin estimada	Actions
04/11/2021	Cancha 2	lincs	un equipo ft	12:00	23:00	

Figura 87: Pantalla de agregar partido a evento

Una vez cargados los partidos, el organizador deberá clicar el botón "Guardar".

Al clicar en el botón “Agregar árbitros”, el sistema mostrará al organizador un modal con un formulario que deberá completar con todos los árbitros a los que quiere que se le envíe una solicitud de participación. Para buscar un árbitro, solo deberá ingresar el email con el que se registró en el sistema. Luego deberá presionar el botón enviar y el sistema enviará una solicitud de participación a todos los árbitros que había seleccionado.

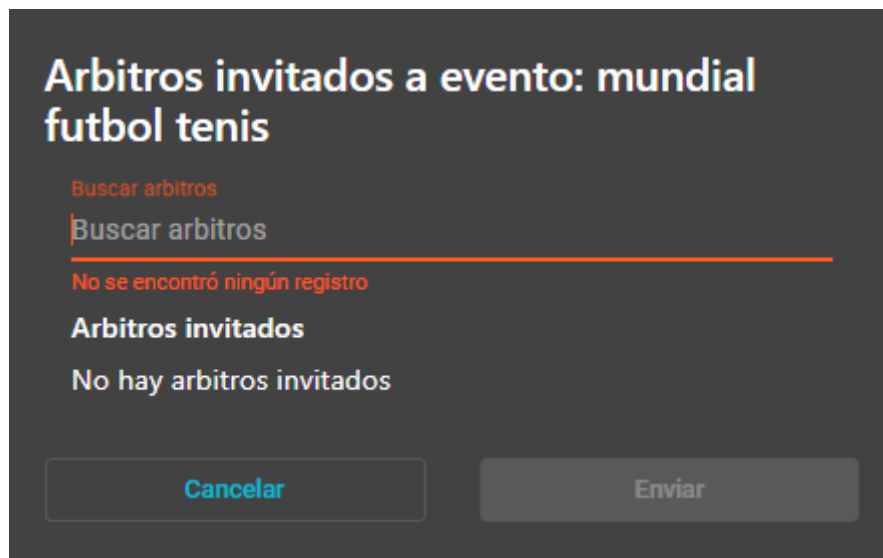


Figura 88: Ventana de invitación de árbitros al evento

### Eliminar Árbitros

Al clicar en el botón “Eliminar árbitros”, el sistema mostrará al organizador un modal con los árbitros participantes del evento, donde podrá eliminar los árbitros que no desee en su evento. Para esto deberá clicar en la cruz que se encuentra al lado del email del árbitro que desea eliminar y finalmente presionar el botón “Guardar cambios”.

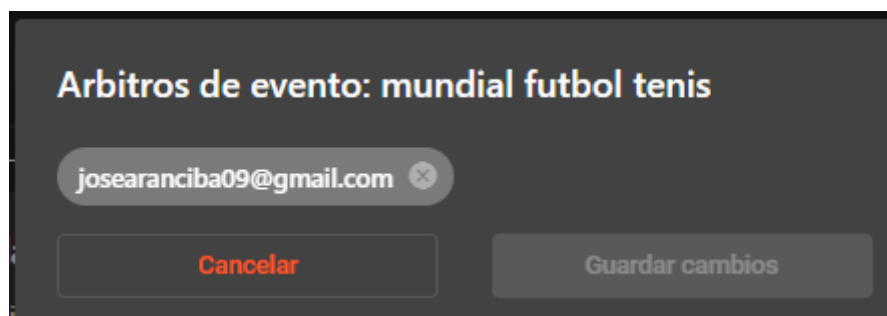


Figura 89: Ventana para eliminar árbitros del evento

### Eliminar Equipos

Al clicar en el botón “Eliminar Equipos”, el sistema mostrará al organizador un modal con los equipos participantes del evento, donde podrá eliminar los equipos que no desee en su

evento. Para esto deberá clicar en la cruz al lado de cada equipo y finalmente en el botón “Guardar cambios”.

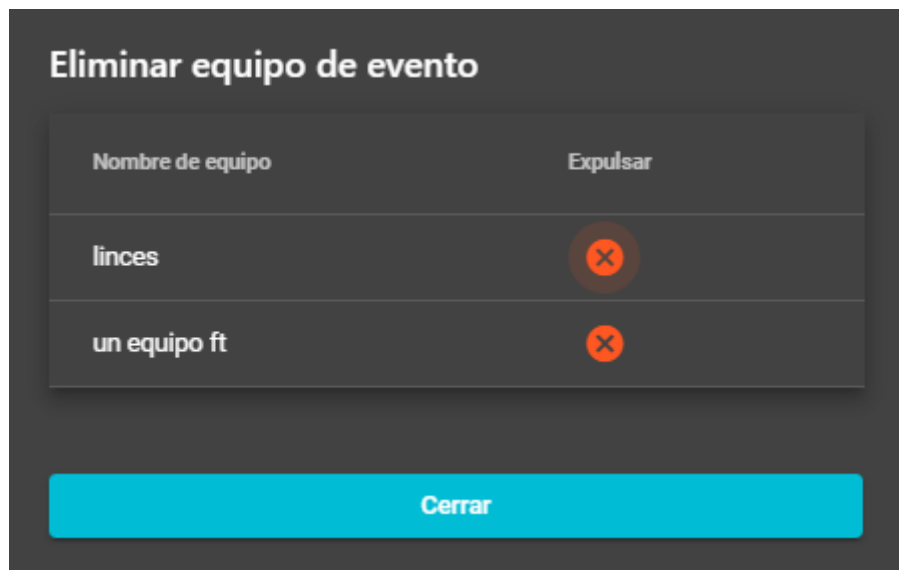


Figura 90: Ventana de eliminación de equipos del evento

### Registrar pagos

Al clicar en el botón “Registrar Pagos”, el sistema mostrará al organizador un modal con los equipos participantes del evento que no han pagado. En el mismo, se podrá registrar el pago de un equipo, presionando el icono “Tick”, en cuyo caso se le cambiará el estado del equipo a “Pagado”. En el caso de querer registrar que un equipo no pago y expulsarlo del sistema, el organizador deberá presionar el icono con forma de cruz.

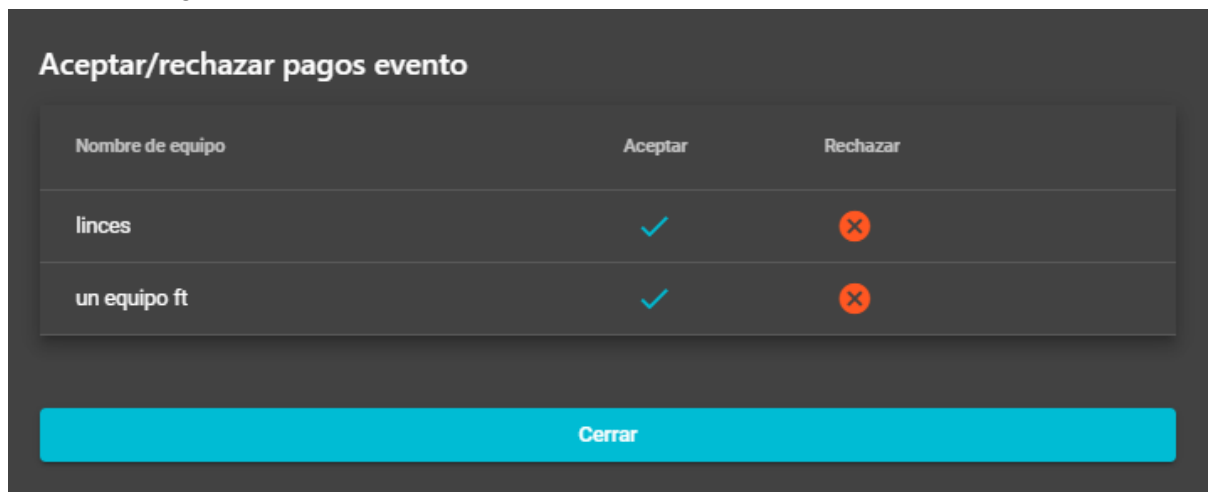




Figura 91: Ventana de registro de pagos de equipos en un evento

*Ver solicitudes de patrocinio*

Al cliquear en el botón de visualización de solicitudes de patrocinio, el sistema mostrará al organizador un modal con todas las solicitudes de patrocinio que el organizador puede aceptar o rechazar. En el mismo, el organizador puede presionar el botón con forma de tick para aceptar la solicitud de patrocinio o el botón con forma de cruz en caso contrario.

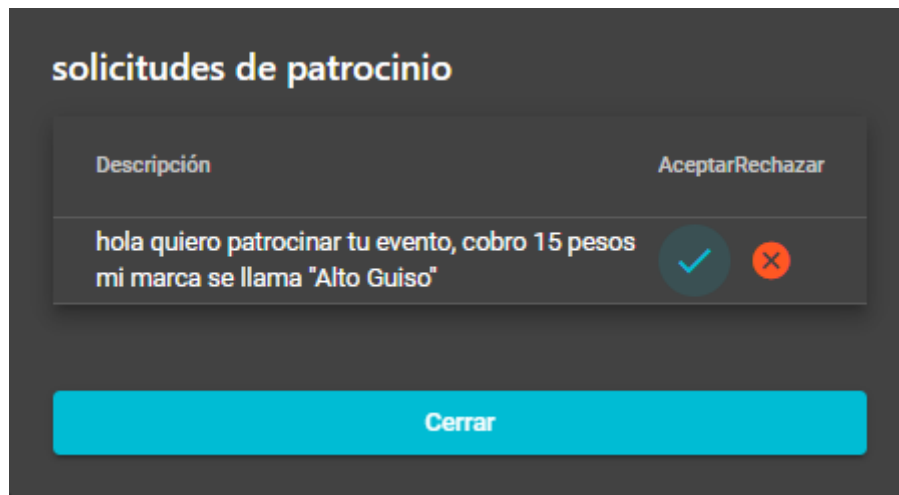


Figura 92: Ventana de solicitudes entrantes de patrocinio

*Visualizar brackets*

El organizador puede visualizar los brackets del evento, que permiten visualizar de una manera muy amistosa, los partidos y los resultados de los mismos. Para esto, el usuario dentro de la pantalla del evento en modo organización, debe dirigirse hasta la sección de brackets, donde podrá visualizarlos.

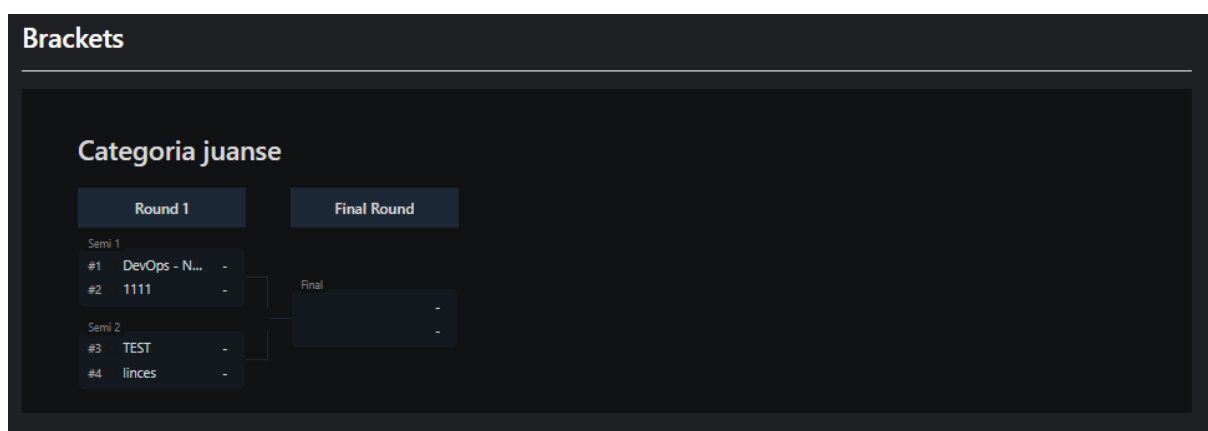


Figura 93: Sección de Brackets

Registrar Resultado de un partido

El organizador puede registrar el resultado de un partido, haciendo clic en el bracket de los equipos cuyo resultado quiere registrar. Al hacerlo, el sistema le mostrará una ventana con un formulario donde deberá ingresar el resultado de cada uno de los equipos y clicar el botón “registrar resultado”.

The screenshot shows a dark-themed form titled "Registrar resultado del equipo: DevOps - Nation". Below the title, there is a label "Ingrese resultado del equipo" and a text input field containing the number "1". To the right of the input field is a small square button with a double-headed arrow. Below this, the form title changes to "Registrar resultado del equipo: 1111", and the label "Ingrese resultado del equipo" is followed by a text input field containing the number "0". At the bottom of the form, there are two buttons: a light blue "Cancelar" button and a dark blue "Registrar resultado" button.

Figura 94: Ventana para registrar resultado de un partido

Tipos de organizadores

Los organizadores tiene la posibilidad de elegir distintos tipos de cuentas con distintos beneficios. Para esto el organizador deberá presionar el botón “Ver cuentas premium” en el sidebar de la izquierda. Esto lo redirigirá a una pantalla donde se le mostrarán todos los tipos de cuentas disponibles, con sus diferentes funcionalidades y precios. Una vez que el usuario elige el tipo de cuenta que desea, solo debe clicar en el botón “Comprar cuenta” y realizar el pago del servicio con Mercado Pago.

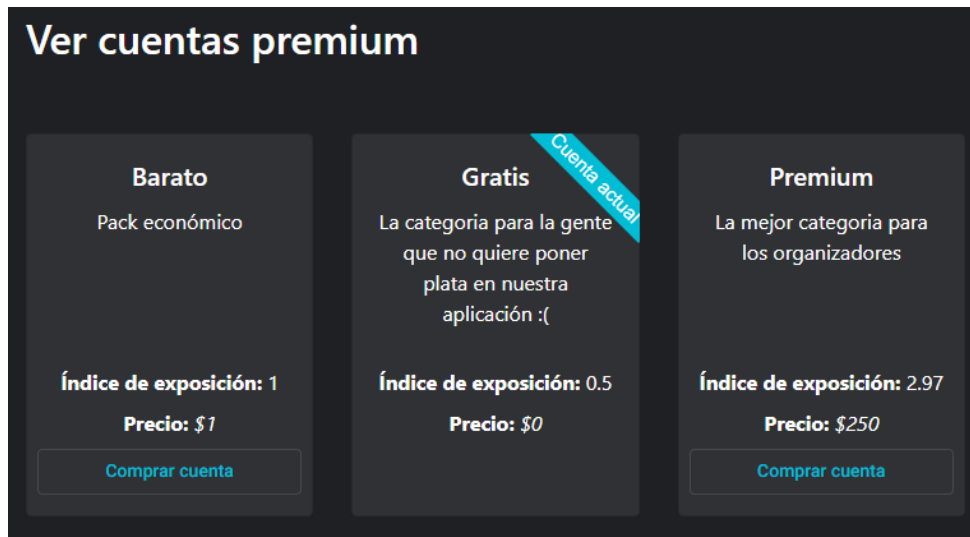


Figura 95: Pantalla de visualización de cuentas de los organizadores

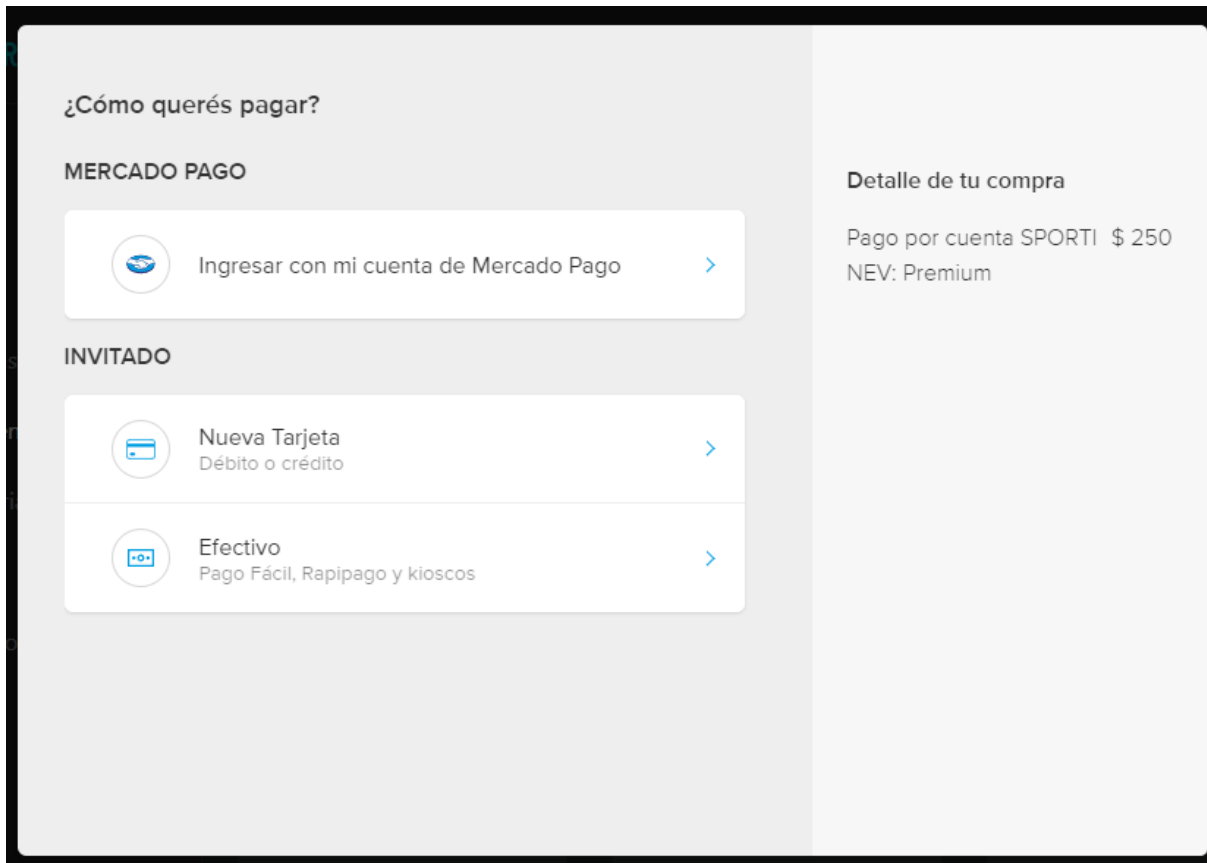


Figura 96: Pantalla de visualización de la integración con Mercado Pago

Vincular cuenta con mercado pago

Los organizadores pueden ofrecerle a los deportistas la posibilidad de pagar el evento mediante Mercado Pago. Pero primero deben tener vinculada su cuenta de Mercado Pago a

su perfil de Sportinev. Para esto el usuario deberá iniciar sesión como organizador, ir a las configuraciones de su perfil y presionar el botón “Vincular cuenta de Mercado Pago”. Esto lo redirigirá a la página oficial de Mercado Pago, donde deberá iniciar sesión y aceptar que Sportinev tenga acceso a su cuenta. Una vez realizado estos pasos, se lo devolverá a la página de Sportinev y ya podrá empezar a crear eventos que acepten como medio de pago Mercado Pago.



Figura 97: Apartado para vincular la cuenta de mercado pago como organizador

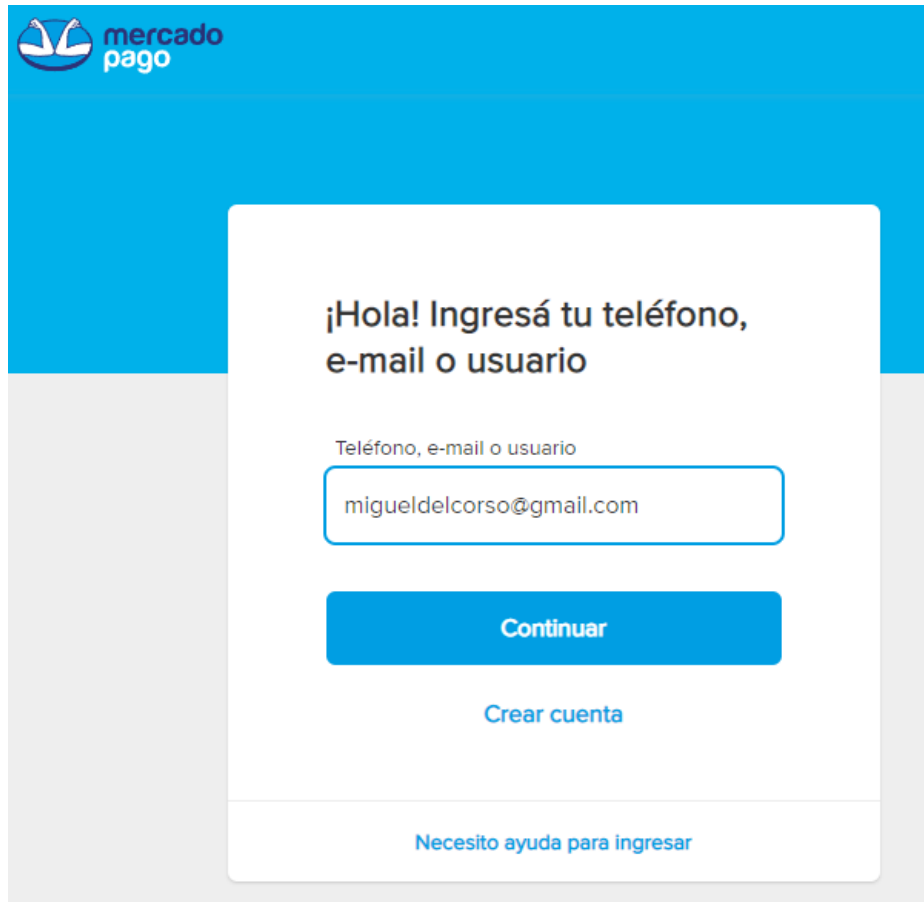


Figura 98: Ingreso a mercado pago con nuestra cuenta.

## Reportes

En el menú lateral de cada usuario organizador, es posible acceder al apartado de reportes, el cual tiene varias opciones de visualización de reportes del sistema. Los posibles reportes son:

### *Tipos de deportes usados para eventos*

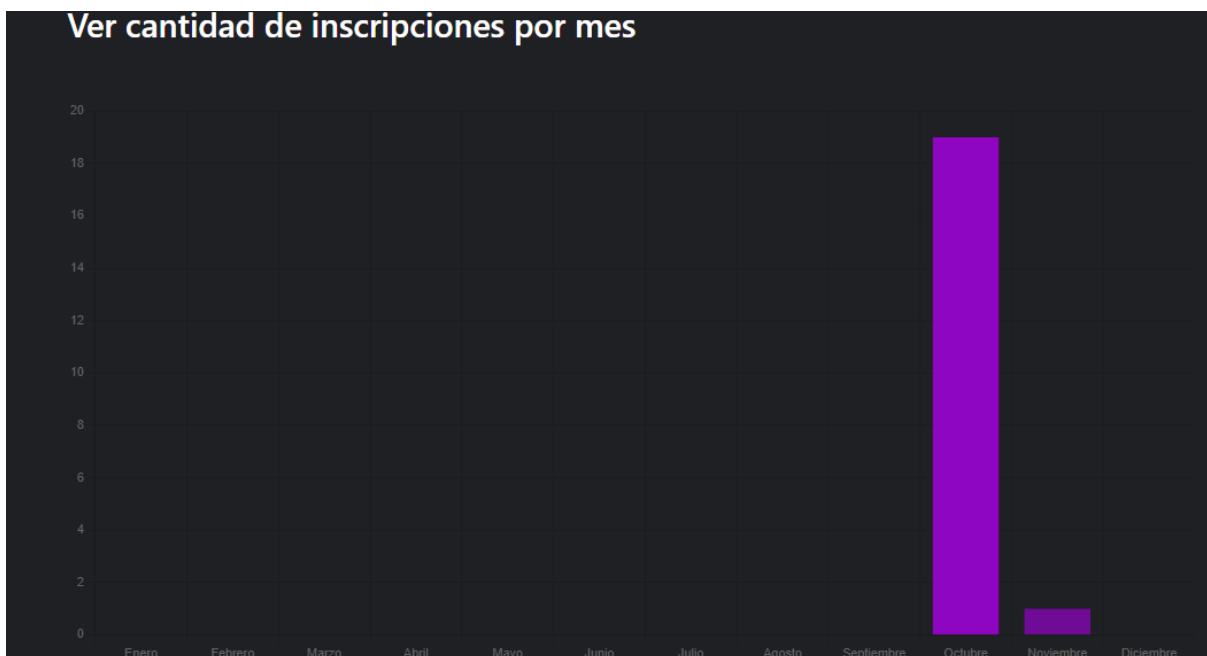
Al cliquear en la pestaña “tipos de deportes usados para eventos”, el usuario organizador es redirigido a una pantalla en donde podrá visualizar la cantidad de eventos que ha creado, separados por el deporte al que pertenecen en forma de diagrama de barras.



Figura 99: Pantalla de reporte de deportes usados para eventos

*Cantidad de inscriptos por mes*

Al cliquear en la pestaña “cantidad de inscriptos por mes”, el organizador es redirigido a una pantalla en donde podrá visualizar la cantidad de deportistas que se inscribieron a sus eventos los meses anteriores.



### Sidebar patrocinador

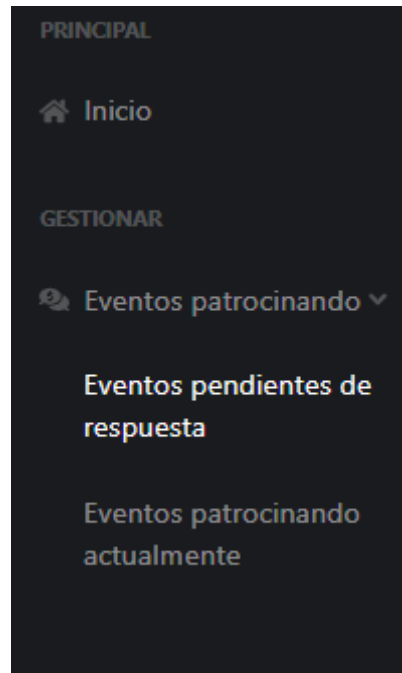


Figura 101: Sidebar del patrocinador

Un usuario con rol “patrocinador” puede realizar una petición de patrocinio en un evento en particular. Para esto, el usuario debe haber iniciado sesión como patrocinador y se debe haber ubicado en la página de visualización de eventos del sistema. Una vez que decide qué evento quiere patrocinar, ingresa al evento cliqueando en “ver más”, y finalmente deberá presionar el botón “Promocionar” en la portada del evento.



Figura 102: Pantalla de visualización de evento patrocinadores

Una vez que el patrocinador cliquea en el botón “patrocinar”, se le mostrará un formulario donde debe completar un mensaje de solicitud de patrocinio explicando los términos del mismo como se observa en la imagen.

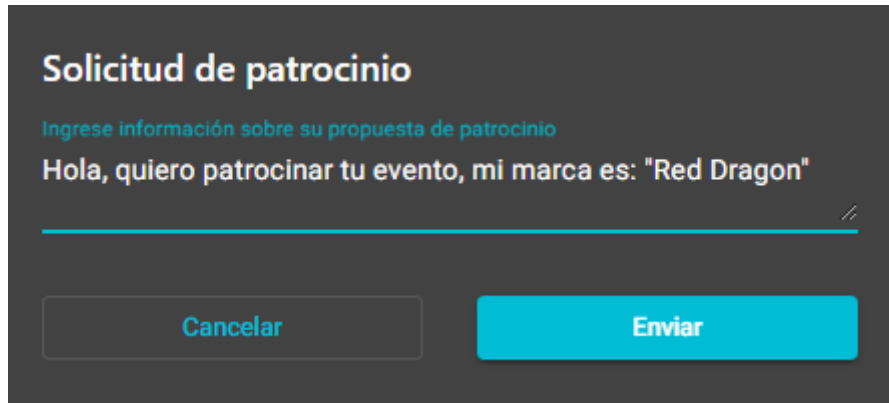
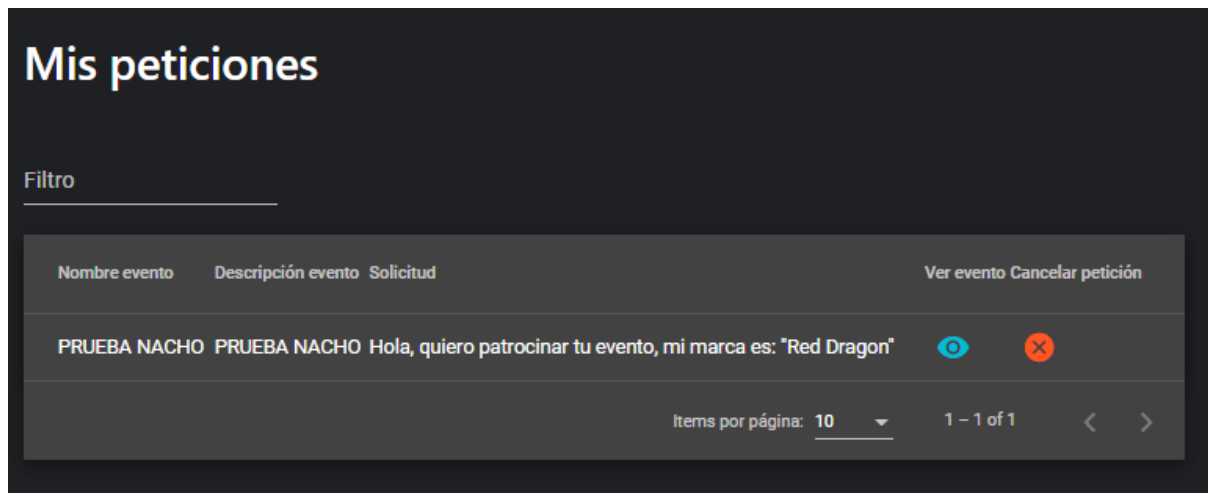


Figura 103: Formulario de solicitud de patrocinio

Quando el usuario completa el formulario, debe clicar el botón “enviar”, para realizar la solicitud de patrocinio y enviarla al organizador dueño del evento.

Visualización de solicitudes de patrocinio

El patrocinador puede observar las solicitudes de patrocinio que hizo, presionando el botón “Ver eventos pendientes de respuesta” en el sidebar de la izquierda. Se redirigirá al usuario a una pantalla donde se mostrarán las solicitudes de patrocinio enviadas por el patrocinador que todavía no han sido respondidas.





Nombre evento	Descripción evento	Solicitud	Ver evento	Cancelar petición
PRUEBA NACHO	PRUEBA NACHO	Hola, quiero patrocinar tu evento, mi marca es: "Red Dragon"		

Figura 104: Visualización de eventos pendientes de respuesta de patrocinio.

Visualización de eventos patrocinados

El patrocinador puede ver los eventos que está patrocinados actualmente, presionando el botón “Eventos promocionando actualmente”, en el sidebar de la izquierda. Se redirigirá al usuario a una pantalla donde se mostrarán los eventos que ha patrocinado y que se encuentran en curso.



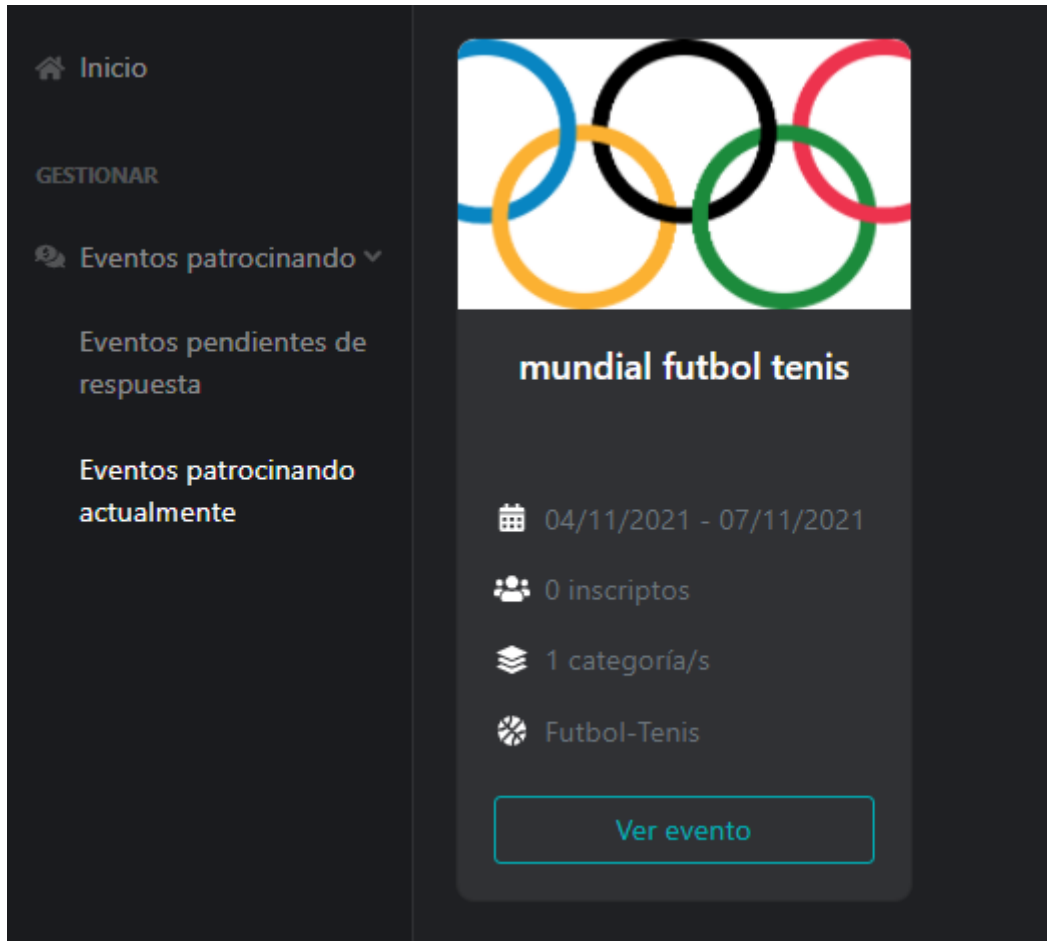


Figura 105: Visualización de eventos patrocinados actualmente

Sidebar árbitro

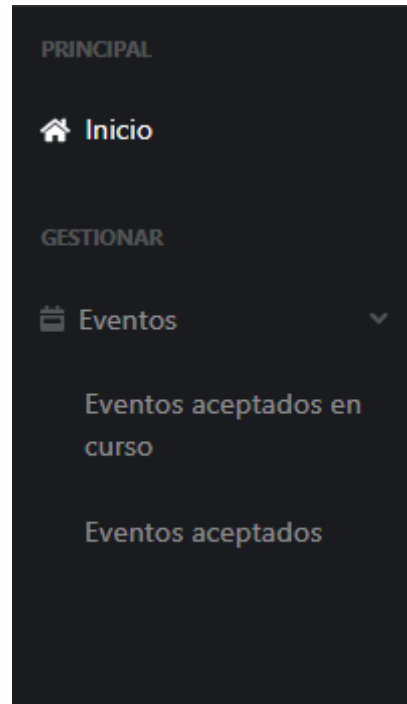


Figura 106: Sidebar del árbitro

Aceptar invitación de arbitraje por mail

Para aceptar una invitación a participar como árbitro en un evento, el usuario deberá buscar en su email la petición de participación que realizó el organizador y luego deberá clicar en el link adjunto.



Figura 107: Email de solicitud de participación como arbitro

Una vez que el árbitro clikea el link del mail para participar en un evento, el usuario es redirigido a la página del evento en donde se le mostrará un cartel para aceptar o rechazar la solicitud de arbitraje del evento.



Figura 108: Ventana de confirmación de participación en evento como árbitro

#### Visualizar eventos aceptados

El árbitro puede visualizar los eventos que aceptó arbitrar cliqueando el botón “Eventos aceptados” en el sidebar de la izquierda. Se redirigirá al usuario a una pantalla donde se le mostrarán solo aquellos eventos en los que él participará.

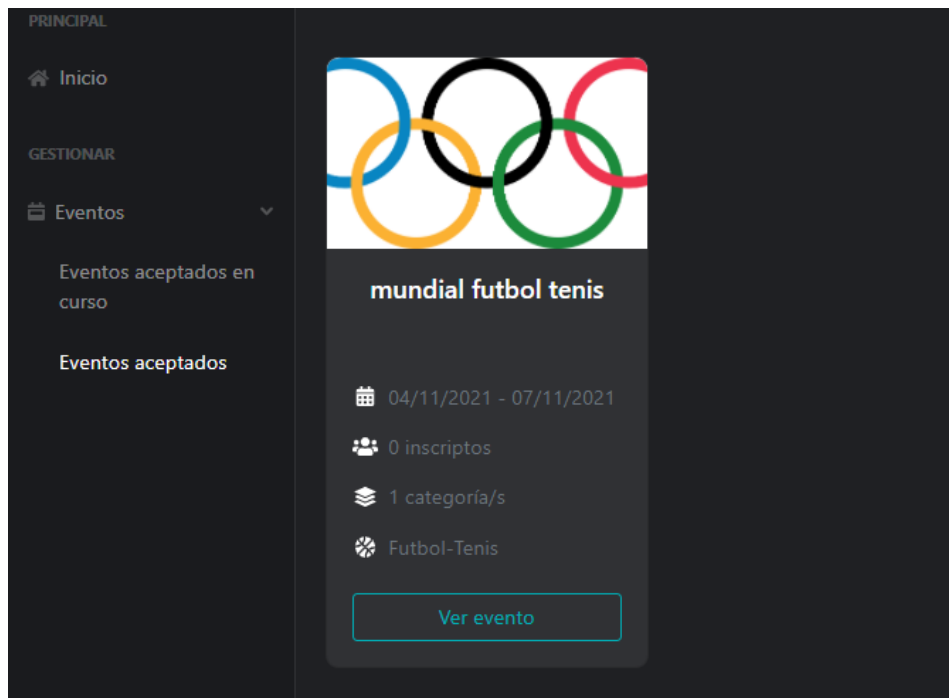


Figura 109: Visualizar eventos aceptados como árbitro

### Visualizar eventos aceptados

De la misma manera en que puede ver los eventos aceptados, el árbitro puede visualizar los eventos que aceptó arbitrar cliqueando el botón “Eventos aceptados en curso” en el sidebar de la izquierda. Se redirigirá al usuario a una pantalla donde se le mostrarán solo aquellos eventos en los que él participa y que se encuentran en curso.

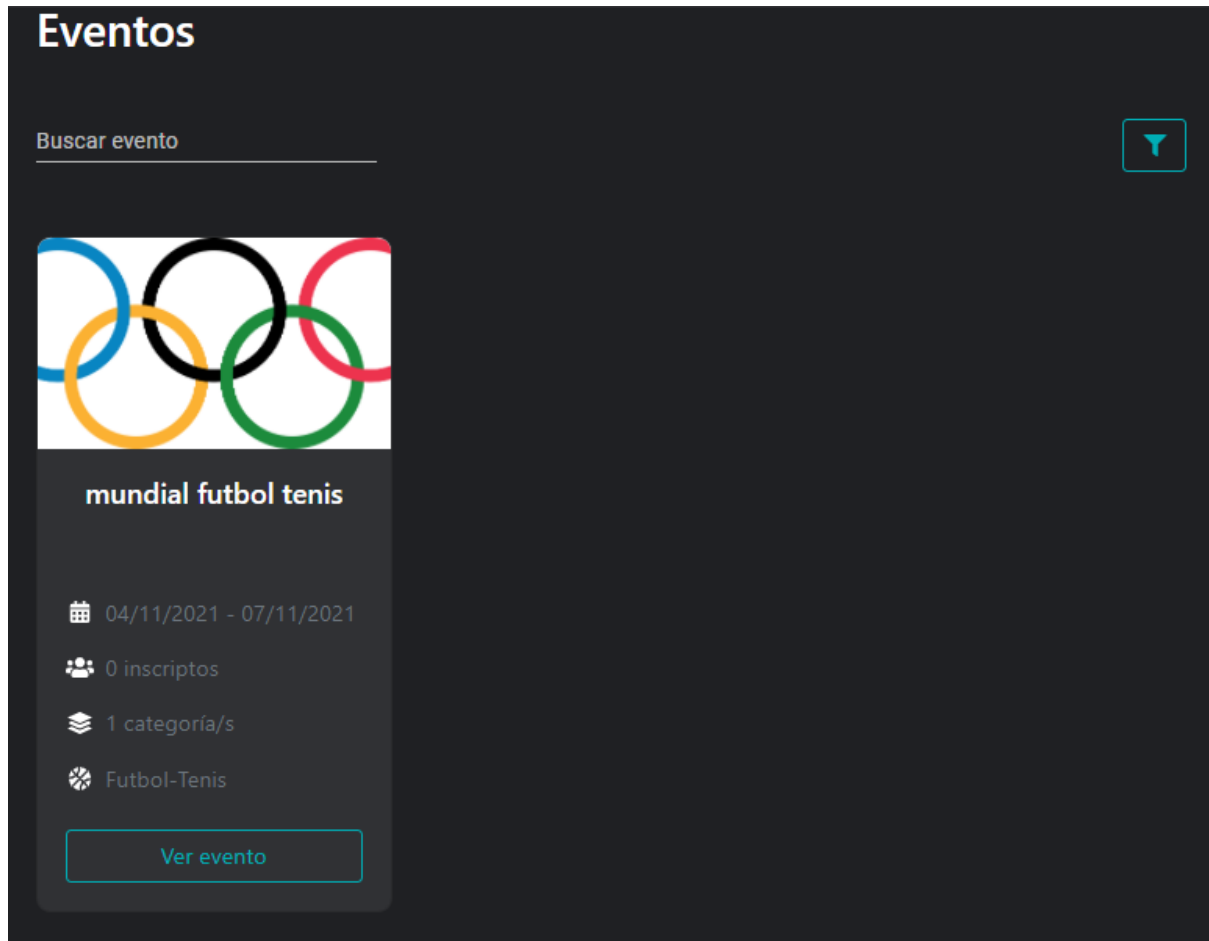


Figura 110: Visualizar eventos aceptados en curso como árbitro

### Sidebar club

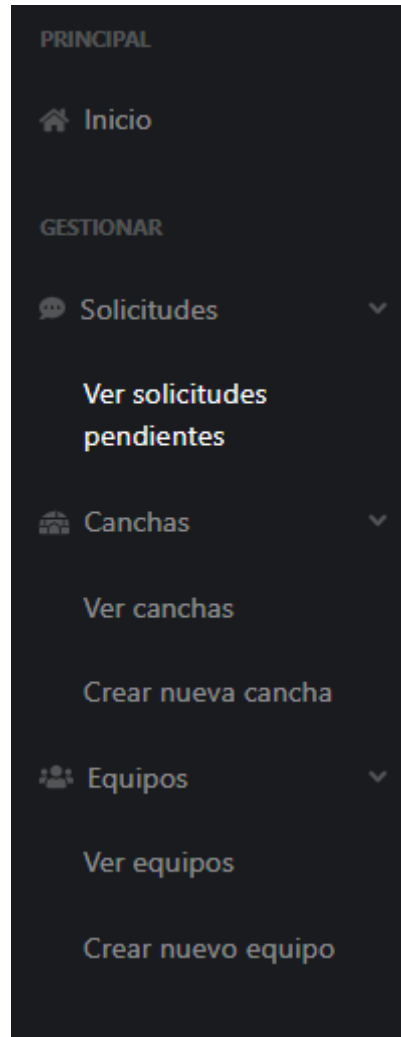


Figura 111: Sidebar del club

#### Visualización de canchas

Para visualizar las canchas como club, el usuario con este rol deberá clicar el botón “Ver canchas”, en el menú lateral:

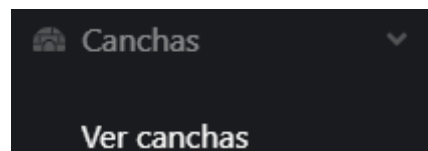


Figura 112: Pestaña de visualización de canchas como club

Luego, el sistema le mostrará una tabla con todas las canchas del club con sus datos:

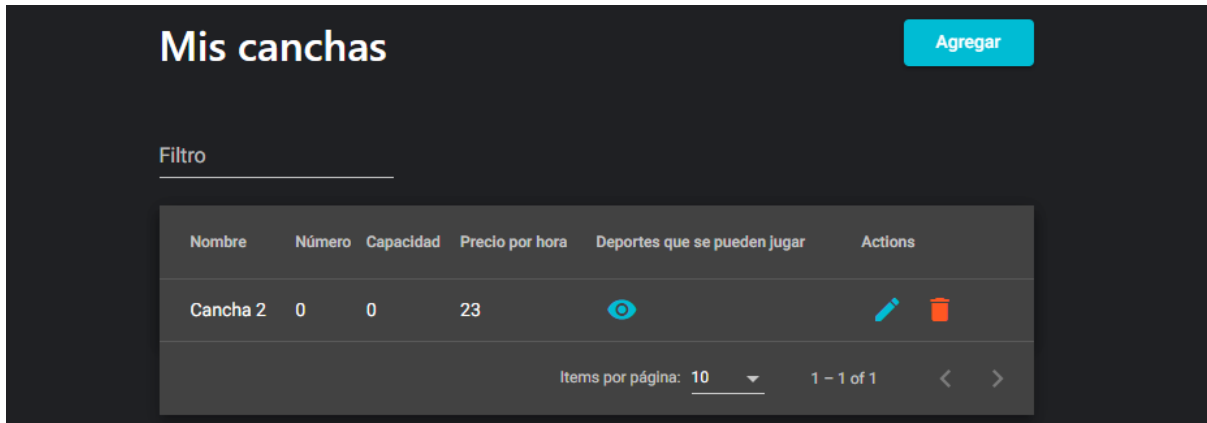


Figura 113: Pantalla de visualización de canchas como club

### Creación y modificación de canchas

Para crear canchas o modificarlas es necesario completar el formulario que se muestra a continuación con datos válidos. En el caso de querer crear una nueva cancha se deberá presionar el botón “Crear nueva cancha” del sidebar de la izquierda. En cambio, si se desea editar una cancha existente, se deberá presionar el icono de lápiz de la fila cuya cancha se desea editar. En cualquier caso el usuario deberá completar el siguiente formulario.

Figura 114: Pantalla de creación de canchas como club

Finalmente, una vez ingresados los datos, debe hacer clic en el botón “Guardar” para guardar la cancha con sus datos en el sistema y poder visualizar la información actualizada en la pantalla de visualización de canchas.

### Visualización de equipos

Para visualizar los equipos como club, el usuario con este rol deberá clicar el botón “Ver equipos”, en el menú lateral:

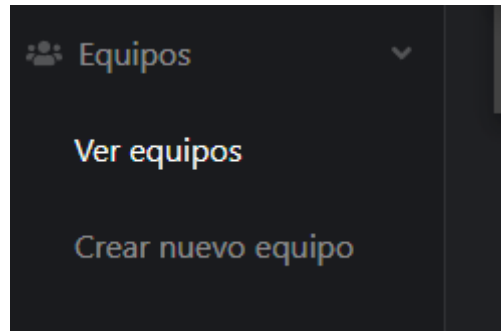
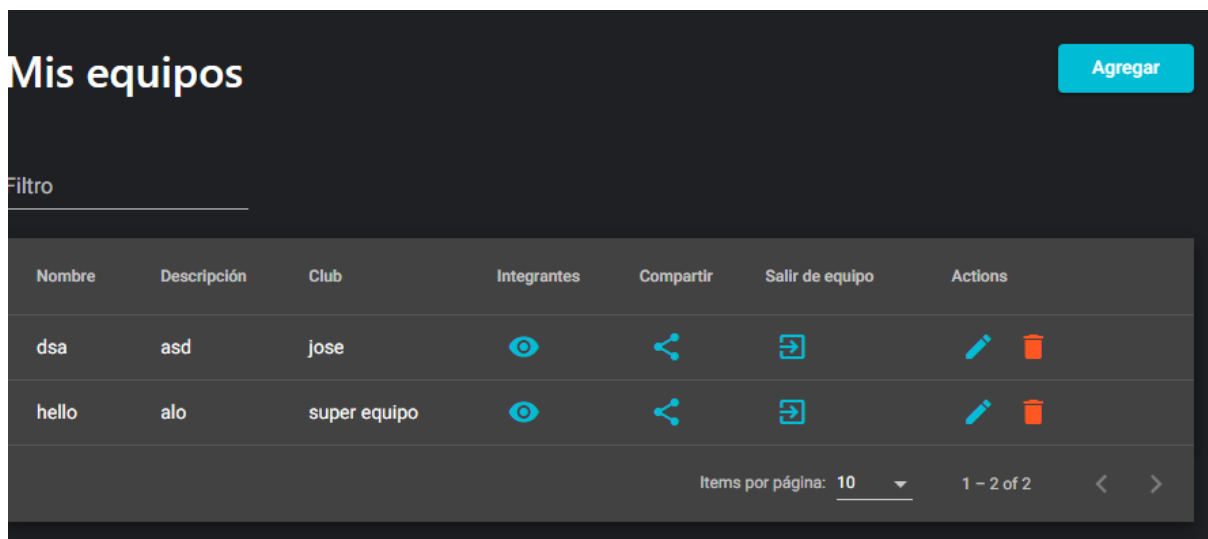


Figura 115: Pestaña de visualización de equipos como club

Luego, el sistema le mostrará una tabla con todos los equipos del club con sus datos:



Nombre	Descripción	Club	Integrantes	Compartir	Salir de equipo	Actions
dsa	asd	jose				
hello	alo	super equipo				


Items por página: 10 1 - 2 of 2

Figura 116: Visualización de equipos como club

### Creación y modificación de equipos

Para crear equipos o modificarlos es necesario completar el formulario que se muestra a continuación con datos válidos. En el caso de querer crear un nuevo equipo se deberá presionar el botón “Crear nueva equipo” del sidebar de la izquierda. En cambio, si se desea editar un equipo existente, se deberá presionar el icono de lápiz de la fila cuyo equipo se desea editar. En cualquier caso el usuario deberá completar el siguiente formulario.

## Equipo



Ingrese el nombre del equipo \*

Real Madrid

Ingrese una descripción del equipo \*

El club mas grande del mundo

### Seleccionar integrantes \*

Buscar integrante

elvio.aruta98@gmail.com

Integrantes seleccionados

- deportista@mozej.com
- jopivax850@ecofreon.com
- elvio.aruta98@gmail.com

[Volver](#) [Guardar](#)

Figura 117: Pantalla de creación de equipos como club

Finalmente, una vez ingresados los datos, debe hacer clic en el botón “Guardar” para los datos en el sistema y poder visualizar la información actualizada en la pantalla de visualización de equipos.

Ver solicitudes de creación de eventos pendientes

Cuando un organizador decide crear un evento en las canchas de un club, el mismo deberá aceptar participar del mismo o el evento se cancelara. Tanto para aceptar la participación como para rechazarla, el usuario deberá iniciar sesión en el sistema y deberá presionar el botón “Ver solicitudes pendientes” del sidebar de la izquierda. Esto lo redirigirá a una pantalla donde podrá visualizar las solicitudes de participación de los eventos y allí mismo podrá realizar aceptarlas o rechazarlas.



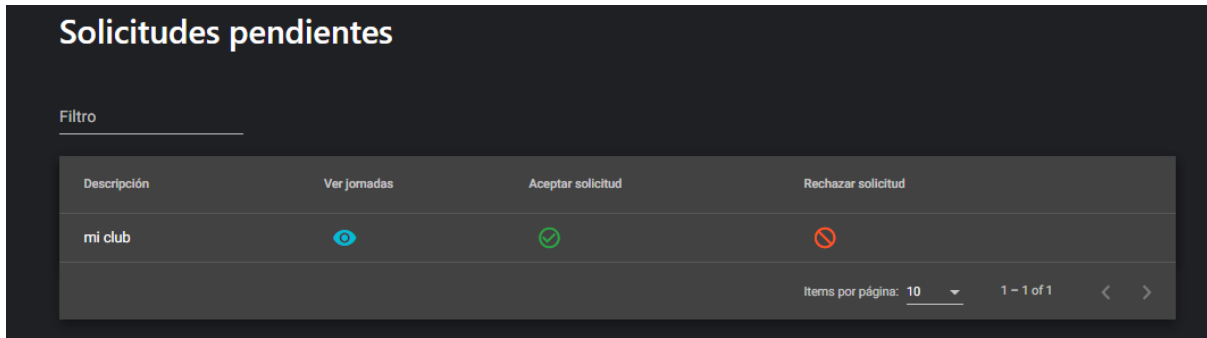


Figura 118: Solicitudes pendientes de club

Una vez clickeado en cualquiera de las acciones, aparecerá un modal con un mensaje de confirmación, y luego de confirmar, la acción tomará efecto, aceptando o rechazando la solicitud de creación.

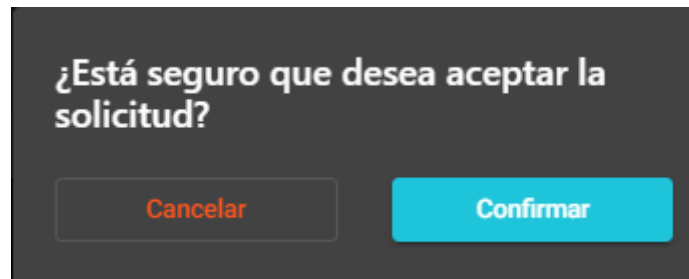


Figura 119: Modal confirmar acción solicitud

## Manejo y visualización de errores

El usuario, al tratar de realizar una acción de forma incorrecta, obtendrá un error por parte del sistema. Estos errores están representados de varias maneras, dependiendo del momento en el que se detecta el error y la gravedad del mismo.

Los errores que se detectan mientras se van ingresando los datos, se representan mediante el cambio a color rojo del campo donde se produjo el error y un mensaje explicando el motivo por el que se produjo. Por ejemplo, en caso de ingresar un email inválido a la hora de crear un nuevo usuario, obtendremos un mensaje de error como el siguiente.



Figura 120: Pantalla de visualización de errores de ingreso de datos

Existen infinidad de situaciones en las que pueden aparecer estos errores, es importante que se lea con detenimiento el mensaje para poder saber por qué se está produciendo el mismo.

Sin embargo, muchos errores no se producen por el ingreso de un dato erróneo, sino por otros motivos que no pueden ser detectados mientras se van ingresando. Es por esto que se deben utilizar otras formas de visualizar los errores. Para los errores que no son muy graves, se usa un pequeño cuadro rojo arriba a la derecha con el mensaje de porque se ha producido el error. Estos errores son muy comunes y pueden surgir por varias situaciones, es fundamental leer el mensaje para poder corregirlo. Por ejemplo, al tratar de acceder al sistema con un email y una contraseña incorrecta, obtendremos un error como el siguiente.



Figura 121:Pantalla de visualización de error de login incorrecto

Para poder ingresar al sistema el usuario deberá cambiar el email y la contraseña que habían ingresado por datos correctos.

Finalmente, en el caso de producirse un error grave, se mostrará un cuadro en el medio de la pantalla y un mensaje de porque se ha producido. Estos mensajes son muy importante para el usuario, por lo que no se cerrarán hasta que se presione el botón "Aceptar". Esta clase de errores no son usuales, pero son muy relevantes, pueden producirse por realizar una acción indebida o por un error no esperado. Por ejemplo, en caso de querer editar un evento que ha finalizado, el sistema mostrará un error como el siguiente.

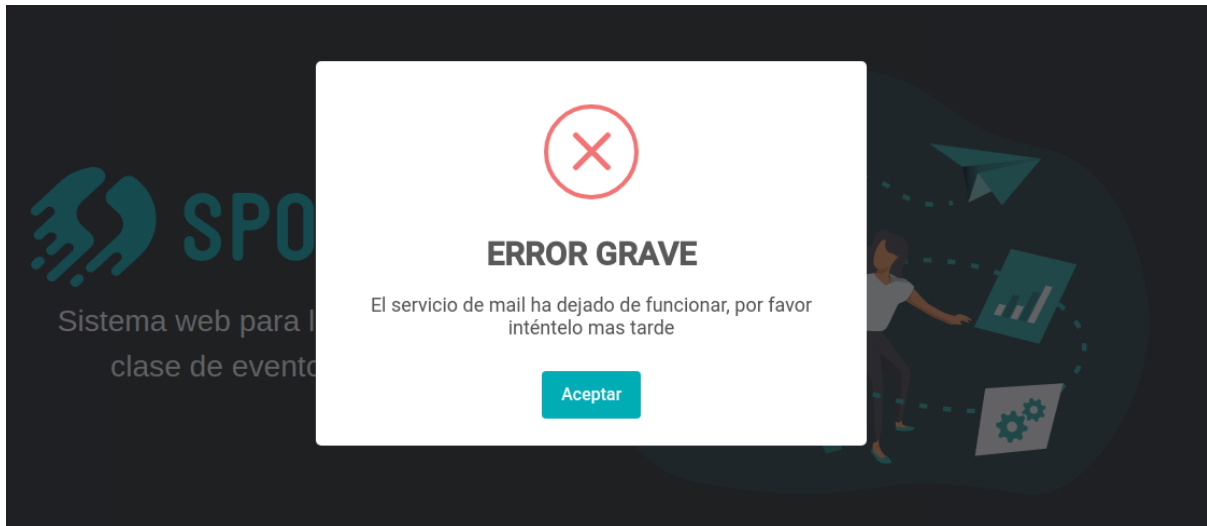


Figura 122:Pantalla de visualización de error grave al querer editar evento finalizado.

El sistema está diseñado para que los mensajes de error sean entendibles y se puedan corregir de forma sencilla, pero en caso de que no se comprenda un mensaje, el usuario podrá contactarse con los analistas del sistema mediante la página de contacto.





**Por favor complete el siguiente cuestionario para que nuestros analistas lo revisen**

Ingrese su nombre

Ingrese su apellido

Ingrese su teléfono

Ingrese su email

Mensaje para administrador

Figura 123:Pantalla de contacto.